GLIF

Thu Feb 17 11:30:05 PST 2000

Package *Temporal_Expression_Package*

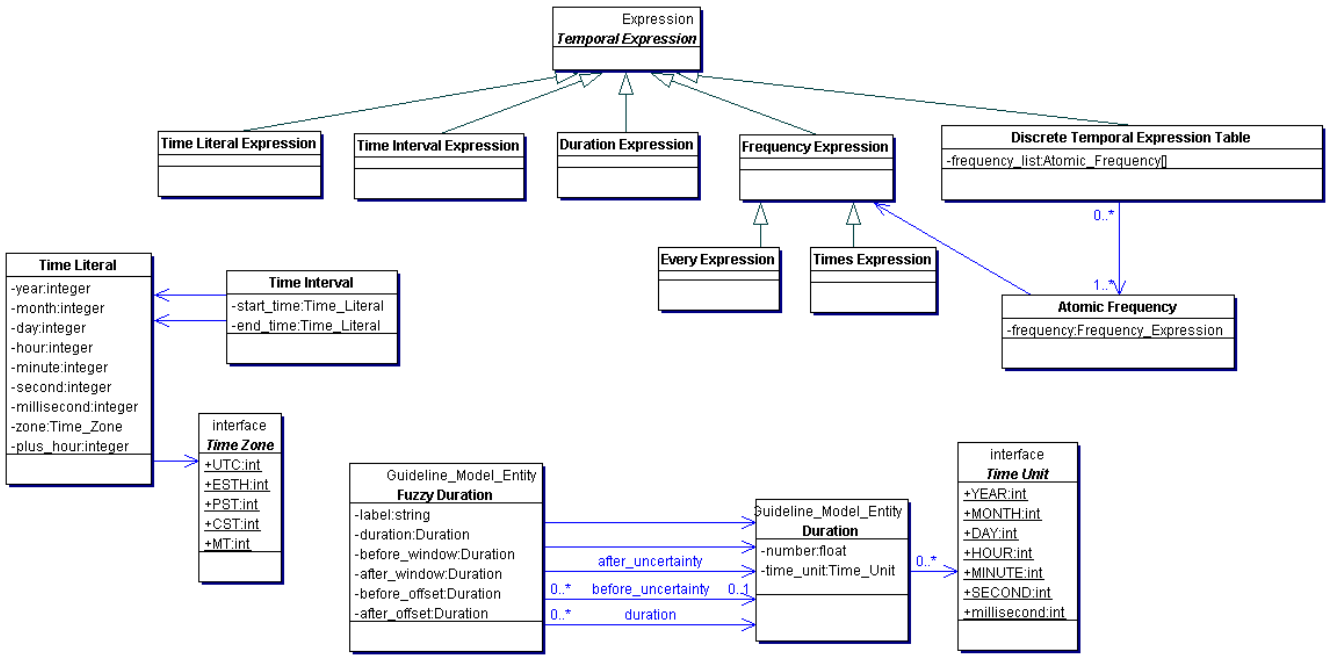
Classes

class Temporal_Expression_Package.Atomic_Frequency
class Temporal_Expression_Package.Discrete_Temporal_Expression_Table
class Temporal_Expression_Package.Duration
class Temporal_Expression_Package.Duration_Expression
class Temporal_Expression_Package.Every_Expression
class Temporal_Expression_Package.Frequency_Expression
class Temporal_Expression_Package.Fuzzy_Duration
class Temporal_Expression_Package.Temporal_Expression
class Temporal_Expression_Package.Time_Interval
class Temporal_Expression_Package.Time_Interval_Expression
class Temporal_Expression_Package.Time_Literal
class Temporal_Expression_Package.Time_Literal_Expression
class Temporal_Expression_Package.Times_Expression

Interfaces

interface Temporal_Expression_Package.Time_Unit
interface Temporal_Expression_Package.Time_Zone

Class Diagram



Interface Node Detail

Interface *Temporal_Expression_Package.Time_Unit*

An enumerated type that enumerates different possible time units

Interface *Temporal_Expression_Package.Time_Zone*

Class Detail

Class *Temporal_Expression_Package.Atomic_Frequency*

Description:
An *atomic frequency* specifies the frequency at which something should occur and the duration after which all iterations should end

Purpose

Considerations:

Example:

Attributes

frequency

Attribute Detail

frequency

Data type: Fuzzy_Duration
Multiplicity: 1
Description:The frequency of the atomic frequency.
Level: B and C

Class *Temporal_Expression_Package.Discrete_Temporal_Expression_Table*

Inherits from:

Temporal_Expression_Package.Temporal_Expression

Description:
A *discrete temporal expression table* specifies a table who consist of rows of pairs of frequecies and durations (the pairs are Atomic frequencies). The order of rows is important. The rows are to be executed from the top of the table to its bottom.

Purpose
To model a list of discrete frequencies.

Considerations:

Example:
see patient every 5 weeks for 5 months, then, every 2 weeks for 1 month, and then every week for 1 month.

Attributes

frequency_list

Attribute Detail

frequency_list

Data type:Atomic_Frequency
Multiplicity: 1:*
Description:The frequency table is a table of discrete frequencies and thier related duration
Level: B and C

Class *Temporal_Expression_Package.Duration*

Inherits from:

Guideline_Model_Entity

Description:
A *duration* has a value and a time unit

Purpose
To store a time value and its time unit

Considerations:

Example:
2 hour

Attributes

number
time_unit

Attribute Detail

number

Data type: float
Multiplicity: 1
Description:the value of the Duration
Level: A, B and C

time_unit

Data type: Time_Unit
Multiplicity: 0..1
Description:the time unit
Level: A, B and C

Class *Temporal_Expression_Package.Duration_Expression*

Inherits from:

Temporal_Expression_Package.Temporal_Expression

Class *Temporal_Expression_Package.Every_Expression*

Inherits from:

Temporal_Expression_Package.Frequency_Expression

Description:
An *every expression*specifies that something should occur every fuzzy duration with an allowed before and after offset

Purpose
In order to represent tight temporal expressions, as shown in the example

Considerations:
We decided to comment the slot "repeat_every" of type Fuzzy_Duration, which is the structured way to specify an every expression, since we alreaqdy have the specification slot (of type string) that will store the duration expression.

Example:
Do something every 8 hours.

Class *Temporal_Expression_Package.Frequency_Expression*

Inherits from:

Temporal_Expression_Package.Temporal_Expression

Description:
A *tight or loose temporal expression* .

Purpose
In order to represent either a tight or a loose temporal expressions

Considerations:
We decided to comment the slot "for" of type Duration, which is the structured way to specify a duration expression, since we alreaqdy have the specification slot (of type string) that will store the duration expression.

Example:

Class *Temporal_Expression_Package.Fuzzy_Duration*

Inherits from:

Guideline_Model_Entity

Description:
A *fuzzy duration* is a duration that has an asociated before and after uncertainty. Any time point within (duration-berfore_uncertainty, duration+after_uncertainty) is considered to be within the duration of the fuzzy duration.

Purpose:
Sometimes a duration is not enough for our modeling purposes, and we need a fuzzy duration.

Considerations::
Example:
The patient should come back to the clinic after 2 weeks, with a before and after uncertainties of 2 days.

Attributes

Attribute Detail

after_offset

Data type: Time_Magnitude
Multiplicity: 0:1
Description:The thing that is being iterated should iterate every "every" duration. If something occurred outside the "every" duration then it is NOT on time. If the occurrence happend in the interval (every.duration - every.before_uncertainty - allowed_before_offset, every.duration - every.before_uncertainty) then the iteration point should be reset to the time of the event occurrence
Level: B and C

after_window

Data type: Time_Magnitude
Multiplicity: 0..1
Description:any time point within (duration, duration+after_uncertainty) is considered to be valid
Level: B and C

before_offset

Data type: Time_Magnitude
Multiplicity: 0:1
Description:The thing that is being iterated should iterate every "every" duration. If something occurred outside the "every" duration then it is NOT on time. If the occurrence happend in the interval (every.duration - every.before_uncertainty - allowed_before_offset, every.duration - every.before_uncertainty) then the iteration point should be reset to the time of the event occurrence
Level: B and C

before_window

Data type: Time_Magnitude
Multiplicity: 0..1
Description:any time point within (duration-before_uncertainty, duration) is considered to be valid
Level: B and C

duration

Data type: Time_Magnitude
Multiplicity: 1
Description:the basic unfuzzy duration
Level: B and C

label

Data type: string
Multiplicity: 0:1
Description:a text representation of the Iteration_Specification
Level: A

Class *Temporal_Expression_Package.Temporal_Expression*

Inherits from:

Expression_Package.Expression

Description:
temporal expressions concepts to be expressed within a guideline. A temporal expression consists of a label, a specification, and didactics inherited from the expression class

Purpose

Considerations:
Temporal expressions which are a function of the number of the iteration, or the time that passed from the first iteration, or the time that passed from some milestone, are represented only as discrete set temporal criterion, and not as a function such as 2i or 2(10-t).

In order for GLIF to include temporal expressions, there has to be a notion of time-stamped events or episodes and time-stamped clinical data/patient-related concepts and this currently does not exist. This could be captured by having variables that have associated start and end times to represent different events/episodes, etc. An instantaneous event would have equal start and end times and an episode with a duration would have an end time greater than the start time. Currently there is no means of analyzing temporal trends in GLIF, and it is probably something we want to support in future versions of GLIF. To facilitate analysis of temporal trends in the future, we could introduce (now) the Arden method of creating a list that describes an event or episode's history with associated start and end time-stamps (e.g., a list of blood pressure readings for a patient at different instances in time where start_time=end_time for each reading).

Example:
every 8 hours
3 times a day

TemporalExpression:
TimeLiteral | Duration | TimeInterval | Frequency | FrequencySet | Duration UnaryOperator | UnaryOperator Number of Identifier | UnaryOperator Identifier | TemporalExpression BinaryOperator TemporalExpression | Identifier BinaryOperator TemporalExpression | TemporalExpression BinaryOperator Identifier | TimeInterval lasting Duration to Duration

Class *Temporal_Expression_Package.Times_Expression*

Inherits from:

Temporal_Expression_Package.Frequency_Expression

Description:
A *times expression* specifies that something should occur a specified number of times within a specified interval (the per attribute)

Purpose
In order to represent loose temporal expressions, as shown in the example

Considerations:
1) This class should not be used during execution. It should be mapped, or refined to a tight temporal expression.
2) We decided to comment the slots "per" of type Duration, and "repeat_times" of type integer, which is the structured way to specify a times expression, since we already have the specification slot (of type string) that will store the duration expression.

Example:
Do something 3 times a day.

Class *Temporal_Expression_Package.Time_Interval*

Attributes

Attribute Detail

end_time

Data type: Time_Literal
Multiplicity: 1
Description:
Level: A

start_time

Data type: Time_Literal
Multiplicity: 1
Description:
Level: A

Class *Temporal_Expression_Package.Time_Interval_Expression*

Inherits from:

Temporal_Expression_Package.Temporal_Expression

); a time-interval_expression specifies an interval which is defined by a start_time and an end_time

Class *Temporal_Expression_Package.Time_Literal*

Attributes

day
hour
millisecond
minute
month
plus_hour
second
year
zone

Attribute Detail

day

hour

millisecond

minute

month

plus_hour

second

year

zone

Class *Temporal_Expression_Package.Time_Literal_Expression*

Inherits from:

Temporal_Expression_Package.Temporal_Expression

a time-literal_expression involves a specific instance in time (expressed as yyyy-mm-ddThh:mm:ss(Z|+/-hh) based on Arden syntax notation

Interface Detail

Interface *Temporal_Expression_Package.Time_Unit*

An enumerated type that enumerates different possible time units

Attributes

DAY
HOUR
millisecond
MINUTE
MONTH
SECOND
YEAR

Attribute Detail

DAY

HOUR

millisecond

MINUTE

MONTH

SECOND

YEAR

Interface *Temporal_Expression_Package.Time_Zone*

Attributes

CST
ESTH
MT
PST
UTC

Attribute Detail

CST

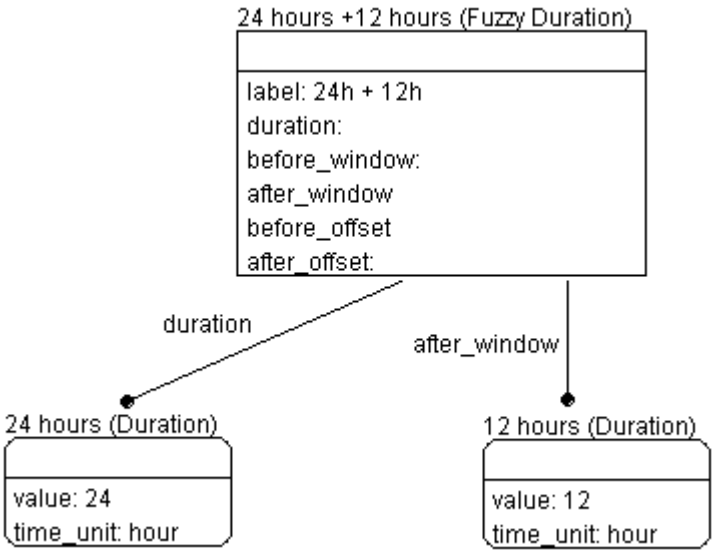
ESTH

MT

PST

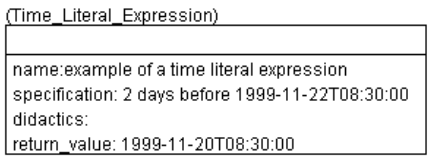
UTC

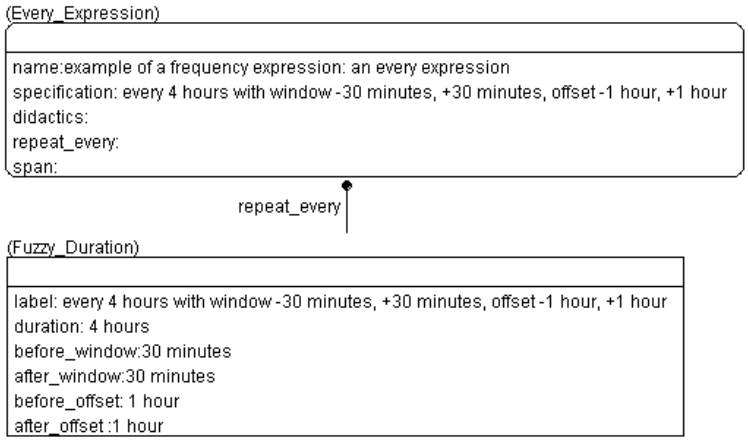
Example:

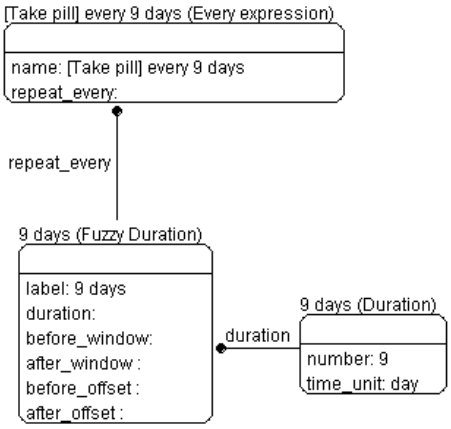


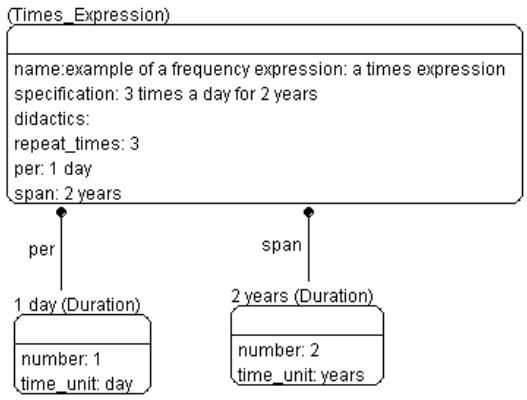
(a) An example of a fuzzy duration. A pill has to be taken every 24 hours. If the person forgot to take the pill, she may take it within 12 hours, otherwise, the dose is skipped.
(b) An example of offsets. A pill should be taken every 4-5 hours. The next pill should be taken 4-5 hours after the previous pill was taken.

Example:


(Time_Literal_Expression)
name:example of a time literal expression
specification: 2 days before 1999-11-22T08:30:00
didactics:
return_value: 1999-11-20T08:30:00


(Every_Expression)
name:example of a frequency expression: an every expression
specification: every 4 hours with window -30 minutes, +30 minutes, offset -1 hour, +1 hour
didactics:
repeat_every:
span:
repeat_every
(Fuzzy_Duration)
label: every 4 hours with window -30 minutes, +30 minutes, offset -1 hour, +1 hour
duration: 4 hours
before_window:30 minutes
after_window:30 minutes
before_offset: 1 hour
after_offset :1 hour


[Take pill] every 9 days (Every expression)
name: [Take pill] every 9 days
repeat_every:
repeat_every
9 days (Fuzzy Duration)
label: 9 days
duration:
before_window:
after_window :
before_offset :
after_offset :
duration
9 days (Duration)
number: 9
time_unit: day


(Times_Expression)
name:example of a frequency expression: a times expression
specification: 3 times a day for 2 years
didactics:
repeat_times: 3
per: 1 day
span: 2 years
per
span
1 day (Duration)
number: 1
time_unit: day
2 years (Duration)
number: 2
time_unit: years

The time-literal expression specification: 2 days BEFORE 1999-11-22T08:30:00 produces the result: 1999-11-20T08:30:00 which is of type time-literal.
The every_expression specification: every 4 hours with window -30 minutes, +30 minutes, offset -1 hour, +1 hour

The times_expression specification: 3 times a day for 2 years