TQ1

Logical View Report

Unified Modeling Language Syntax

Includes Attributes

Includes Documentation

Generated December 22, 1998

1:48:45 AM

C:\Doc\HL7\HIMSS Demo 99\HMDs\PythConvertMETL\TQ1.mdl

TABLE OF CONTENTS

TABLE OF CONTENTS 2

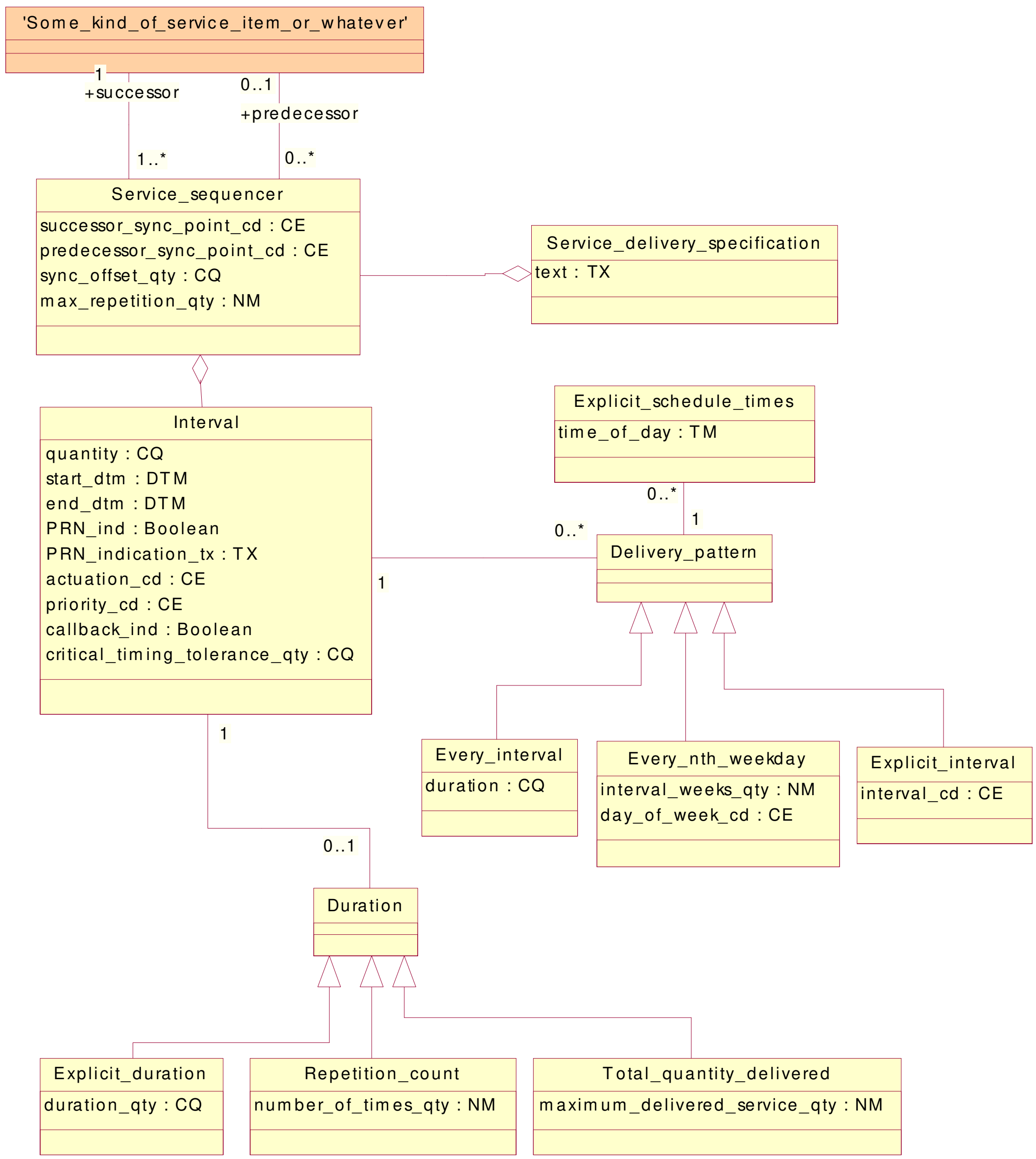
LOGICAL VIEW REPORT 3

LOGICAL VIEW 4
Service_delivery_specification 4
Interval 4
Priority 5
Delivery_pattern 5
Every_interval 6
Every_nth_weekday 6
Explicit_interval 6
Treatment_pattern 6
Explicit_schedule_times 6
Duration 7
Explicit_duration 7
Repetition_count 7
Total_quantity_delivered 7
Service_sequencer 8
'Some_kind_of_service_item_or_whatever' 8

TOTALS: 9

LOGICAL PACKAGE STRUCTURE 9

DIAGRAM



'Some_kind_of_service_item_or_whatever'
1
+successor
0..1
+predecessor
1..*
0..*
Service_sequencer
successor_sync_point_cd : CE
predecessor_sync_point_cd : CE
sync_offset_qty : CQ
max_repetition_qty : NM
Service_delivery_specification
text : TX
Interval
quantity : CQ
start_dtm : DTM
end_dtm : DTM
PRN_ind : Boolean
PRN_indication_tx : TX
actuation_cd : CE
priority_cd : CE
callback_ind : Boolean
critical_timing_tolerance_qty : CQ
Explicit_schedule_times
time_of_day : TM
0..*
1
0..*
1
Delivery_pattern
Every_interval
duration : CQ
Every_nth_weekday
interval_weeks_qty : NM
day_of_week_cd : CE
Explicit_interval
interval_cd : CE
1
0..1
Duration
Explicit_duration
duration_qty : CQ
Repetition_count
number_of_times_qty : NM
Total_quantity_delivered
maximum_delivered_service_qty : NM

LOGICAL VIEW REPORT

Logical View

Service_delivery_specification

A specification of the quanitity, priority, delivery pattern, and duration of potentially repeated deliveries of a single service. One instance of this class is roughly equivalent to one TQ data field in version 2.3.

The specified quanitity, priority, delivery pattern, and duration of the single service can vary through multiple instances of the Interval class.

A combined pattern of delivery of multiple services can be described by combining instances of Service_delivery_specification, each of which is associated with a different service. The way in which the instances of Service_delivery_specification is combined is described by instances of Sequencer.

Private Attributes:

text : TX

This field is a full text version of the instruction (optional).

Issue: what should happen if the text description disagrees with the coded description. Does a value in the text description imply a need for human review? How would such a review be recorded?

Interval

An occurrence of this class describes repeated delivery of a quantity of a single service from a start time to the lesser of an end time or the completion of a duration specification, with the repetition being described by various instances of species of Delivery_pattern.

Variations in the quantity and more elaborate variations in the delivery pattern can be described by using multiple instances of Interval for a single instance of Service_delivery_specification. This is described under the Service_sequencer class.

Private Attributes:

quantity : CQ

The quantity of the service item to be delivered at one time. This could be the number of tablets, quantity of liquid, minutes of exercize, etc.

start_dtm : DTM

This field may be specified by the orderer, in which case it indicates the earliest date/time at which the services should be started. In many cases, however, the start date/time will be implied or will be defined by other fields in the order record (e.g., priority - STAT). In such a case, this field will be empty.

The filling service will often record a value in this field after receipt of the order, however, and compute an end time on the basis of the start date/time for the filling service's internal use.

end_dtm : DTM

When filled in by the requester of the service, this field should be the latest date/time that the service should be performed. If it has not been performed by the specified time, it should not be performed at all. The requester may not always fill in this value, yet the filling service may fill it in on the basis of the instruction it receives and the actual start time.

Regardless of the value of the end date/time, the service should be stopped at the earliest of the date/times specified by either the duration or the end date/time.

PRN_ind : Boolean

If this attribute is true, the service is to be delilvered as required. The delivery pattern need not be present, but if it is present it specifies the most frequently that the service should be delivered, e.g., Tylenol #3, PRN Q4H.

PRN_indication_tx : TX

The indication for the delivery of the PRN service, e.g., for pain.

Issue: although prescrptions are written this way, does this really belong here?

actuation_cd : CE

This code allows one to distinguish between the time and priority at which a service should be actuated (e.g., blood should be drawn) and the time and priority at which a service should be completed (e.g., results should be reported). Values are {actuation, completion}.

Most instances of Interval describe actuation, however when one or more instances of Interval exists for actuation, one may also exist for completion.

priority_cd : CE

This field describes the urgency of the request. The values are: {Stat, With highest priority; ASAP, lower than Stat, higher than Routine, Routine, Preop urgency based on surgery schedule, Timing critical, it is critical to come as close as possible to the requested time, e.g., for a trough antimicrobial level.}

callback_ind : Boolean

If true, the completion of the service should be regarded as stat and notifiction should occur when the service is completed (e.g., call back with the results of this test).

Issue: is this really a characteristic of timing and quantity or does it belong elsewhere in the model?

critical_timing_tolerance_qty : CQ

If priority_cd has the value for critical timing, this optional field may specify the tolerance. The quantity portion of the field may be seconds, minutes, hours, days, weeks, or months.

When appearing in the first sequential instance of Interval, the tolerance is specified relative to the actuation time for the service. When specified in another instance, it refers to the tolerance relative to the predecessor instance.

Priority
Delivery_pattern

The pattern of delivery of the service during an interval. There must be a pattern unless the PRN_ind in Interval is 'true'. There may be at most two instances of Delivery_pattern per instance of Interval.

Two instances of Delivery_pattern are used to express certain nonuniform patterns that correspond to common medical usage, such as QOD BID.

The first pattern specifies the frequency of a major cycle of delivery. Its units specifies the duration of a major cycle. The second pattern describes the frequency of delivery within the major cycle. Examples: Q3D TID(major cycle 3 days, duration one day, deliver the service 3 times on that day at the institutionally established times); Q2M Q2D (major cycle, two months, duration one month, minor cycle every other day, which means 'every other month deliver the service every other day, but in the alternate month don't deliver the service)

More elaborate patterns can be described by multiple repetitions of Interval.

Every_interval

Q<integer><units>, where units can be seconds, minutes, hours, days, weeks, or months, examples Q2week (every second week), Q6H every six hours.)

Derived from Delivery_pattern

Private Attributes:

duration : CQ

Used to specify that the service should be delivered every fixed period of time, e.g, Q6H. The possible values for units are seconds, minutes, hours, days, weeks, or months.

Every_nth_weekday

Used for patterns such as every other Monday, (interval_weeks_qty = 2, day_of_week_cd = the code for Monday.)

Derived from Delivery_pattern

Private Attributes:

interval_weeks_qty : NM
day_of_week_cd : CE

Explicit_interval

An interval code, any of BID, TID, QID, QAM, QSHIFT, QHS, QPM, C, ONCE.

Derived from Delivery_pattern

Private Attributes:

interval_cd : CE

Treatment_pattern

Private Attributes:

PRN_ind : Boolean

Explicit_schedule_times

When present, the times override the institutionally scheduled times that are implicit in the repeat_pattern.

Private Attributes:

time_of_day : TM

Duration

The period of time for which services should be provided for a given instance of Interval. The period of time may be explicitly specified, derived from indirect specifications, or it may be not stated. If there is no instance of Duration associated with an instance of Interval, the service is delivered for an indefinite duration. Delivery of the service may be ended because the end_dttm value in Interval is reached, because the service is discontinued or for other reasons not described in this part of the model.

Explicit_duration

The duration specified in terms of a number and a units which is a period of time, e.g., 3 months.

Derived from Duration

Private Attributes:

duration_qty : CQ

Repetition_count

The duration is derived from the number of occurrences of the unit of the service associated with the associated instance of Interval. For example, if the Service_delivery_specification is associated with a 5 mg tablet of diazepam, the quantity per administration in Interval is 2, number_of_times_qty is 10, and the delivery pattern is QHS, the derived duration would be 10 days. (10 repretitions of the delivery of two tablets, once per day.)

Derived from Duration

Private Attributes:

number_of_times_qty : NM

Total_quantity_delivered

The duration is derived from quanity and frequency of delivery of services during the Interval. For example, if this Service Delivery Specification is associated with 250 mg tablets of ampicillin, quantity two, repetition pattern QID, maximum_delivered_service_qty 1000 the derived duration would be 5 days.

Derived from Duration

Private Attributes:

maximum_delivered_service_qty : NM

See Class description. The units associated with this number is implicitly the same as the units of the service item that is associated with this instance of Service_delivery_specification.

Service_sequencer

This class specifies the sequence and certain other timing relationships between service items.

"Asynchronous" actuation is specified by having several service items that succeed the same predecessor. An example of this might be "prednisone given at 1 tab on Monday, Wednesday, Friday, and at 1/2 tab on Tuesday, Thursday, Saturday, Sunday." Other examples might be orders for an enema QPM, liquid diet at midnight, pre-op shave, and diazepam QHS.

"Synchronous" actuation is specified by having several intervals succeed one another. An example of this might be, "measure blood pressure Q15 minutes for the 1st hour, then every 2 hours for the next day." A synchronous actuation can be tied to several predecessors.

The instances and associations can be used to create cyclic patterns. If Interval occurrence B succeeds occurrence A, and A succeeds occurrence B, then a cycle is created. Certain attributes of this class are intended for use in the specific occurrence that "closes the loop" (i.e., refers to a successor that is also a direct or indirect predecessor.) In many cases it may not be practical to identify any single instance as the one that closes the loop. The definitions for the attributes will be effective anyway.

Service items may have distinct actuation and completion phases, such as drawing the blood and returning the results. Synchronization may be tied to actuation or completion, so that a requesting a consultation might be tied to the receipt of lab and radiology results.

Private Attributes:

successor_sync_point_cd : CE

Used to indicate whether the predecessor should synchronize with the start or end of the successor. Values: {start, end}.

Example: if the value of this attribute is "Start" and the value of predessor_sync_point is "Start", and the sync_offset_qty is +2 hours, then the the successor service should start two hours after the start of the predecessor service.

predecessor_sync_point_cd : CE

Used to indicate whether the successor should synchronize with the start or end of the predecessor. Values: {start, end}.

sync_offset_qty : CQ

A positive or negative number, with the units being seconds, minutes, hours, days, weeks, or months. Example, if this value is +2 hours and the values of successor_sync_point_cd and predecessor_sync_point_cd are both "start", the successor should start 2 hours after the start of the predecessor.

max_repetition_qty : NM

If the instances create a cycle, the process of delivering the services may step through an instance of Service_sequencer more than once. If this optional field has a value, the successor will be actuated no more than the number of times specified in this attribute.

'Some_kind_of_service_item_or_whatever'

Because of the putative notion that TQ is a data type, this "class" is meant to represent anything that could contain an attribute that has the TQ data type.

TOTALS:

2 Logical Packages
15 Classes

LOGICAL PACKAGE STRUCTURE

Logical View