



Subpackages

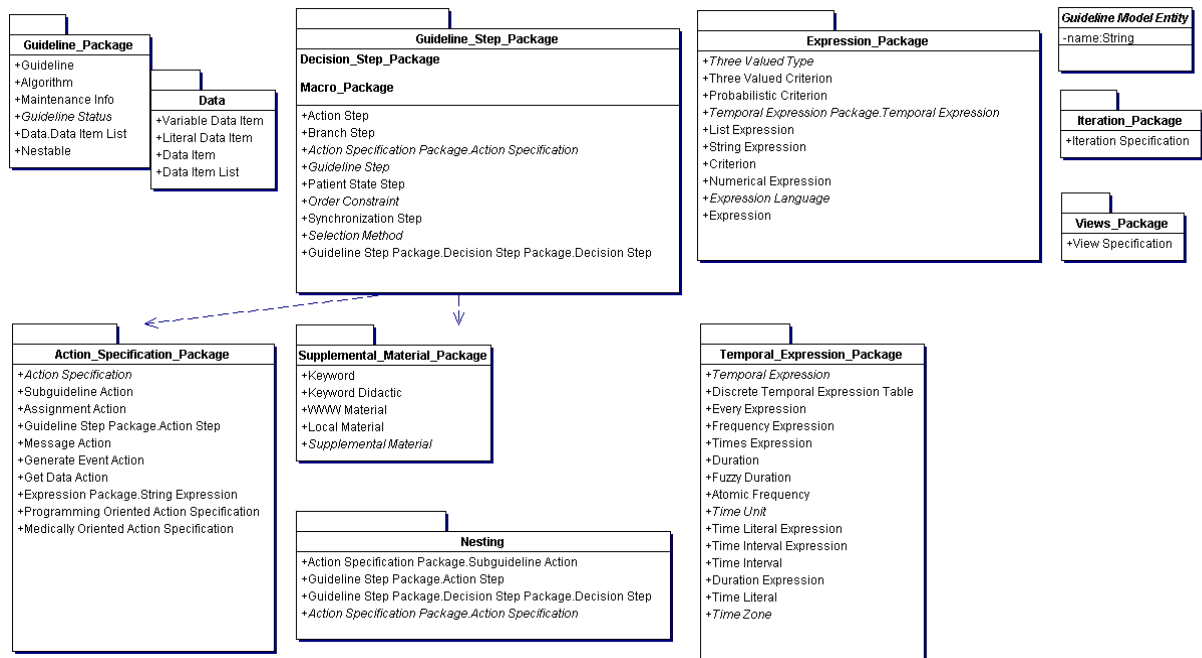
package [Action Specification Package](#)
 package [Data](#)
 package [EventsAndExceptions](#)
 package [Expression Package](#)
 package [Guideline Package](#)
 package [Guideline Step Package](#)
 package [Iteration Package](#)
 package [Nesting](#)
 package [Supplemental Material Package](#)
 package [Temp](#)
 package [Temporal Expression Package](#)
 package [Views Package](#)



Classes

class [Guideline Model Entity](#)

Class Diagram



Package Node Detail



Package [Action Specification Package](#)



Package [Data](#)

Description:

Decision steps direct flow from one *guideline step* to another. A *decision step* may link a *guideline step* to any other *guideline step*. A *decision step* contains a condition, which is an expression. The *condition's* value determines the control flow to one of a

set of possible *guideline steps*, which are specified by the *options* of the *decision step*. The *condition* is compared, using an *operator* to the *options.condition_value*. If the *condition* matches one of the *decision options* then the control can flow, in the case of a choice step, and must flow, in the case of case step, to the *guideline step* that is specified by that *decision option's destination*. If the *condition* does not match any of the set of values specified by the *decision options*, or, if available data do not allow evaluation of the *condition*, then the control flows to the default destination *guideline step*.

When a guideline step finished its execution and the control flow is about to pass to the next step, then, if the next step has associated triggering events, then this next step is executed only after one of its triggering event occurred.

Purpose:

The *conditional step* used an extended Boolean model. This made it cumbersome and error-prone to represent criteria that do not have a true-or-false outcome (e.g., what is the patient's age category: neonate, infant, toddler, child, adolescence, adult, elderly). Therefore, the *case step* replaced the *conditional step* by allowing a conditional choice to be made among several alternative *guideline steps*.

Considerations:

1. The operator was added so that different comparisons would be possible. For example the operator may be "=", or may be "in" to represent a comparison to an interval (age in (30,40) as opposed to age eq > 30)



Package [EventsAndExceptions](#)

Different Event types are defined: end of a previous guideline step, patient arrival, patient data availability, and temporal events, such as a certain point of time has arrived.

Action- and decision steps have an attribute, called *triggering_events*, which specifies what events trigger the start of the step, and with what temporal constraints. The triggering events are of the class *EventStartTimeConstraint*. This class specifies an event and the earliest and latest times after which the step should be started following the occurrence of the triggering event.

Action- and decision steps have an attribute, called *exceptions*, which specifies what exceptions should be checked during the execution of the step. The exceptions are of the class *Exception*. This class specifies the exception-event that should be checked for, a (guarding) condition and a next step. If the exception event occurs and the condition holds, then we terminate the step associated with the exception, and move on to the next step that is defined by the exception.



Package [Expression_Package](#)

The Expression class is a parent Class for all expressions: criterion, temporal expression, list expression, numerical expression and string expression. Different expression languages can be used. Currently, BNF grammar is given for a modification of the Arden Syntax logic grammar, called GLIF_ARDEN. While this gives guideline users the flexibility to specify criteria in a language of their choosing, it is possible that the resulting guidelines will not be sharable because constructs supported by one language (e.g., temporal constructs) cannot be expressed in another language.



Package [Guideline_Package](#)



Package [Guideline_Step_Package](#)

Dependency Links

to ClassDiagram [Action Specification Package](#)

automatic

to ClassDiagram [Supplemental Material Package](#)

automatic



Package [Iteration_Package](#)

The *Iteration_Specification* class specifies information regarding the loop structure of the iteration. The action- and decision steps that should be iterated, that references the *Iteration_Specification*, are iterated until the abort condition or stopping condition criteria hold. The iterations are carried out at a certain (fuzzy) frequency (e.g., every 8h +/- 30min or, 3 times a day). If the iteration does occur within the specified frequency, then the iteration time points remain according to schedule, and are not reset (e.g., the patient came in after 8 1/2 hours. The next iteration should take place after 7 1/2 hours). Some frequencies may have a specified allowed offset. This means that if the iteration began outside the fuzzy frequency, but within the offset, then the

iteration takes place and the iteration points are reset. For example, the patient came back after 9 hours. He gets treated (since the offset is 8 +/-2 h) but the next iteration is 8 hours late.

Package Nesting

Nesting

Nesting of guideline steps is supported in the following manner:

Action Steps are nested by including a Subguideline_Action type of task in the step. The Subguideline_Action task has a subguideline attribute that contains the nested subguideline.

Decision Steps are nested by specifying a subguideline in the decision_detail attribute of the step. This subguideline is executed before the decision criterion for that step is evaluated. The subguideline would modify or create new variables and assign them values. The use of these variables in the decision criteria makes the decision nested.

A nested action or decision can potentially contain any kind of step including other actions, decisions, and other nested steps. Nesting is very useful for managing the complexity of guidelines. Nesting enables looking at a guideline from a top-level view and then zooming into/out of some of its parts. Nesting is also useful in representing a guideline in the context of other guidelines. Since nesting allows grouping of parts of a guideline into a single unit, this is a mechanism that can allow model extensibility and reuse of part of a guideline (defining macros), or adaptation of a guideline to a specific institution by replacing specifications for parts of a guideline (i.e., replacing a goal with a procedure). By use of nesting, steps will be defined as containing other steps. As such, the guideline will be iteratively refined from a coarse level to a fine level of detail.

Package Supplemental Material Package

Package Temporal Expression Package

Package Views Package

Class Detail

Class *Guideline Model Entity*

Attributes

name

Attribute Detail

 **name**