GLIF

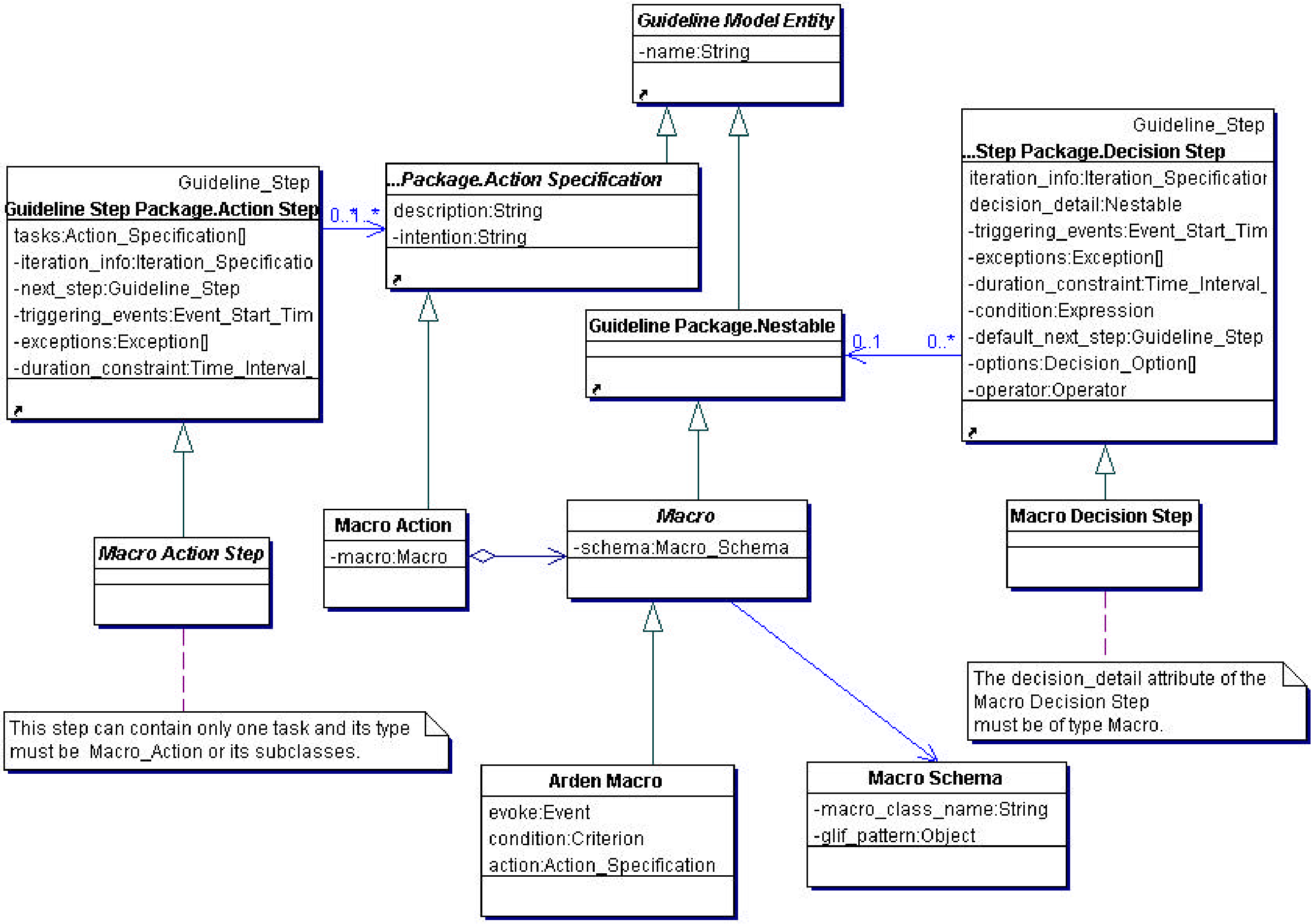
Thu Feb 17 11:30:05 PST 2000

Package *Guideline_Step_Package.Macro_Package*

Classes

class Guideline_Step_Package.Macro_Package.Arden_Macro
class Guideline_Step_Package.Macro_Package.Macro
class Guideline_Step_Package.Macro_Package.Macro_Action
class Guideline_Step_Package.Macro_Package.Macro_Decision_Step
class Guideline_Step_Package.Macro_Package.Macro_Schema
class Guideline_Step_Package.Macro_Package.Macro_Action_Step

Class Diagram



Guideline Model Entity
-name:String
Guideline_Step
..Step Package.Decision Step
iteration_info:Iteration_Specificatior
decision_detail:Nestable
-triggering_events:Event_Start_Tim
-exceptions:Exception[]
-duration_constraint:Time_Interval_
-condition:Expression
-default_next_step:Guideline_Step
-options:Decision_Option[]
-operator:Operator
Guideline_Step
Guideline Step Package.Action Step
tasks:Action_Specification[]
-iteration_info:Iteration_Specificatio
-next_step:Guideline_Step
-triggering_events:Event_Start_Tim
-exceptions:Exception[]
-duration_constraint:Time_Interval_
0..1..*
...Package.Action Specification
description:String
-intention:String
Guideline Package.Nestable
0..1
0..*
Macro Action Step
Macro Action
-macro:Macro
Macro
-schema:Macro_Schema
Macro Decision Step
This step can contain only one task and its type must be Macro_Action or its subclasses.
The decision_detail attribute of the Macro Decision Step must be of type Macro.
Arden Macro
evoke:Event
condition:Criterion
action:Action_Specification
Macro Schema
-macro_class_name:String
-glif_pattern:Object

Note:
This step can contain only one task and its type must be Macro_Action or its subclasses.

Note:
The decision_detail attribute of the Macro Decision Step must be of type Macro.

Class Detail

Class *Guideline_Step_Package.Macro_Package.Arden_Macro*

Inherits from:

Guideline_Step_Package.Macro_Package.Macro

Attributes

action
condition
evoke

Attribute Detail

action

condition

evoke

Class *Guideline_Step_Package.Macro_Package.Macro*

Inherits from:

Guideline_Package.Nestable

Attributes

schema

Attribute Detail

schema

Data type: Macro_Schema
Multiplicity: 1
Description: The schema description of the GLIF pattern and mapping of attributes of the macro to their values in the content_step.
Level: B

Class *Guideline_Step_Package.Macro_Package.Macro_Action*

Inherits from:

Action_Specification_Package.Action_Specification

Attributes

macro

Attribute Detail

macro

Class *Guideline_Step_Package.Macro_Package.Macro_Action_Step*

Inherits from:

Guideline_Step_Package.Action_Step

Description:

Many guidelines or their sections can often be represented using a *pattern* of GLIF steps and components. For example, a clinical trial protocol consists of phases, each phase consists of visits, during a visit, the trial coordinator usually performs certain actions (orders tests, delivers treament) and completes case report forms. A risk assessment guideline consists of data collection steps, followed by risk calculation, followed by recommendations that are based on the risk profile. In fact, even an Arden Syntax MLM , could be described using a pattern of GLIF components: an event (trigger), followed by a decision step (logic slot), followed by action steps (action slot). The pattern of GLIF varies in the description of certain parameters in each of these examples. In the case of the Arden Syntax MLM, these are the actual event, decision, and actions in that instance of the MLM.
A macro step is a (potentially though not necessarily nested) guideline step that describes such patterns of GLIF components in a

custom step. The attributes of the macro step are the parameters of the pattern of GLIF components. The attributes of a *Clinical Trial Visit* macro step are the time of the visit, the tasks to be performed during the visit, the forms to be completed, etc.

Using macro step to represent commonly occuring patterns has benefits for authoring, visual understanding, and execution of guidelines. For example, using macro steps in a protocol authoring tool can make it easier to edit protocols. The author has to deal withconcepts that are at the domain-level (such as visits, forms, etc.) rather than creating flowcharts with action steps, branch steps, and decision steps.

Note that *Macro Step* is an abstract class. Attributes of a particular type or class of Macros would be defined in a subclass of *Macro Step*.

The GLIF pattern of a Macro Step is described in the Macro Schema object that is referenced by the step. This pattern maps attributes of the macro to components in a GLIF step. Thus, using the schema and the values of attributes, a complete sequence of GLIF objects can be constructed. Because the knowledge of the macro can be transformed to a sequence of GLIF components, even applications that are unaware of particular macros and their attributes, can display and execute the underlying GLIF. In this way, macros can be used to create application-specific components with minimal effect on sharability of GLIF.

Purpose:
To develop custom/application-specific extensions of GLIF steps.

Considerations:
The design of an object model for macros has two challenges

- Creating a general macro step that could be distinguished as an action or a decision, while using single inheritance in the class diagram. The current design can distinguish the macro as being an action or a decision by examining the class of the step described in the schema (schema.glif_pattern). Thus, if this step is an action step, the macro is an action. If this step is a decision step, the macro is a decision step.
- We would like to create a formal method to describe the pattern of GLIF components (including the permitted variations) and the mapping of attributes of the macro to their value descriptions in the content step. One possible way to do so, is to use a language similar to the Schema representation of XML. Much work is required to develop such a language. We defer the development of the macro schema language until later. The Macro_Schema class provides a placeholder.

Class *Guideline_Step_Package.Macro_Package.Macro_Decision_Step*

Inherits from:

Guideline_Step_Package.Decision_Step_Package.Decision_Step

Note:
The decision_detail attribute of the Macro Decision Step must be of type Macro.

Class *Guideline_Step_Package.Macro_Package.Macro_Schema*

Description:
A macro schema is an abstract description of a particular class of macros. The schema describes the GLIF structure/pattern of a Macro and the mapping between the GLIF structure and the attributes of the macro.

A GLIF pattern describes the sequence of GLIF steps and other GLIF components that define the contents of the macro.

Purpose:
The schema defines the mapping of a macro to the underlying GLIF. The GLIF pattern is used to generate instances of GLIF objects that can be executed or viewed as such. The schema is also used for validation of macro attributes.

The schema consists of a single action step or decision step. The action step or decision step may contain a nested subguideline of any complexity.

Considerations:
A notation/language for schemas still has to be developed. This language will be used to describe schemas abstractly. The definition of this language will require significant effort, which we will defer until other aspects of GLIF are more fully developed.

We are considering a notation similar to the XML schema definition syntax for the World Wide Web Consortium (W3C).

The examples provided in this document are only an indication of our current ideas on how GLIF schemas will be described. These are not intended to be specifications.

Attributes

glif_pattern
macro_class_name

Attribute Detail

glif_pattern

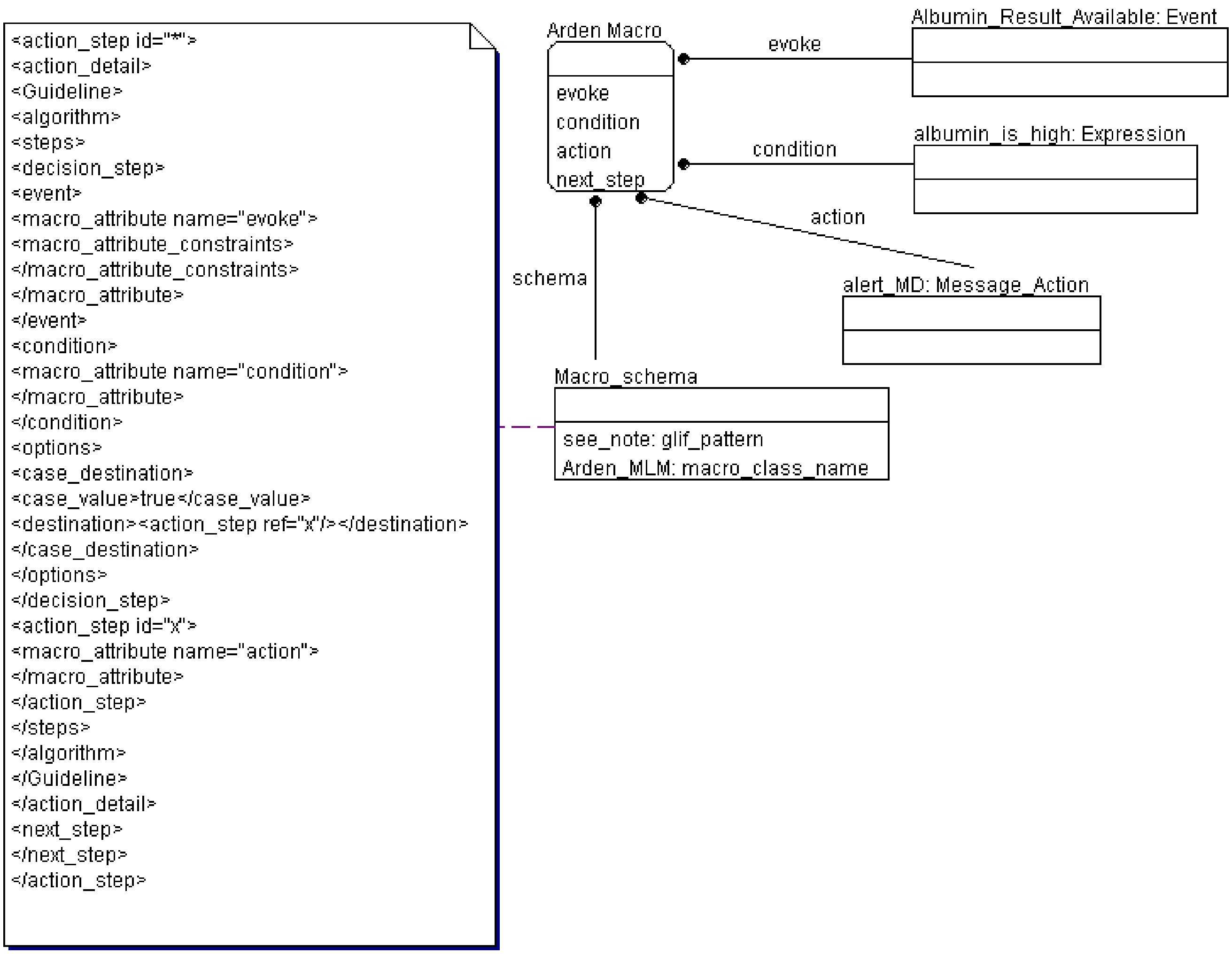
Data type: Object (to be defined)
Multiplicity: 1

Description: The description of the GLIF pattern in a notation to be defined.
Level: B

macro_class_name

Data type: String
Multiplicity: 1
Description: The class of macros that are encoded by this instance of Macro_Schema, e.g., "Clinical Management State" from GEODE.
Level: A

Example:



```
<action_step id="*">
<action_detail>
<Guideline>
<algorithm>
<steps>
<decision_step>
<event>
<macro_attribute name="evoke">
<macro_attribute_constraints>
</macro_attribute_constraints>
</macro_attribute>
</event>
<condition>
<macro_attribute name="condition">
</macro_attribute>
</condition>
<options>
<case_destination>
<case_value>true</case_value>
<destination><action_step ref="x"/></destination>
</case_destination>
</options>
</decision_step>
<action_step id="x">
<macro_attribute name="action">
</macro_attribute>
</action_step>
</steps>
</algorithm>
</Guideline>
</action_detail>
<next_step>
</next_step>
</action_step>
```

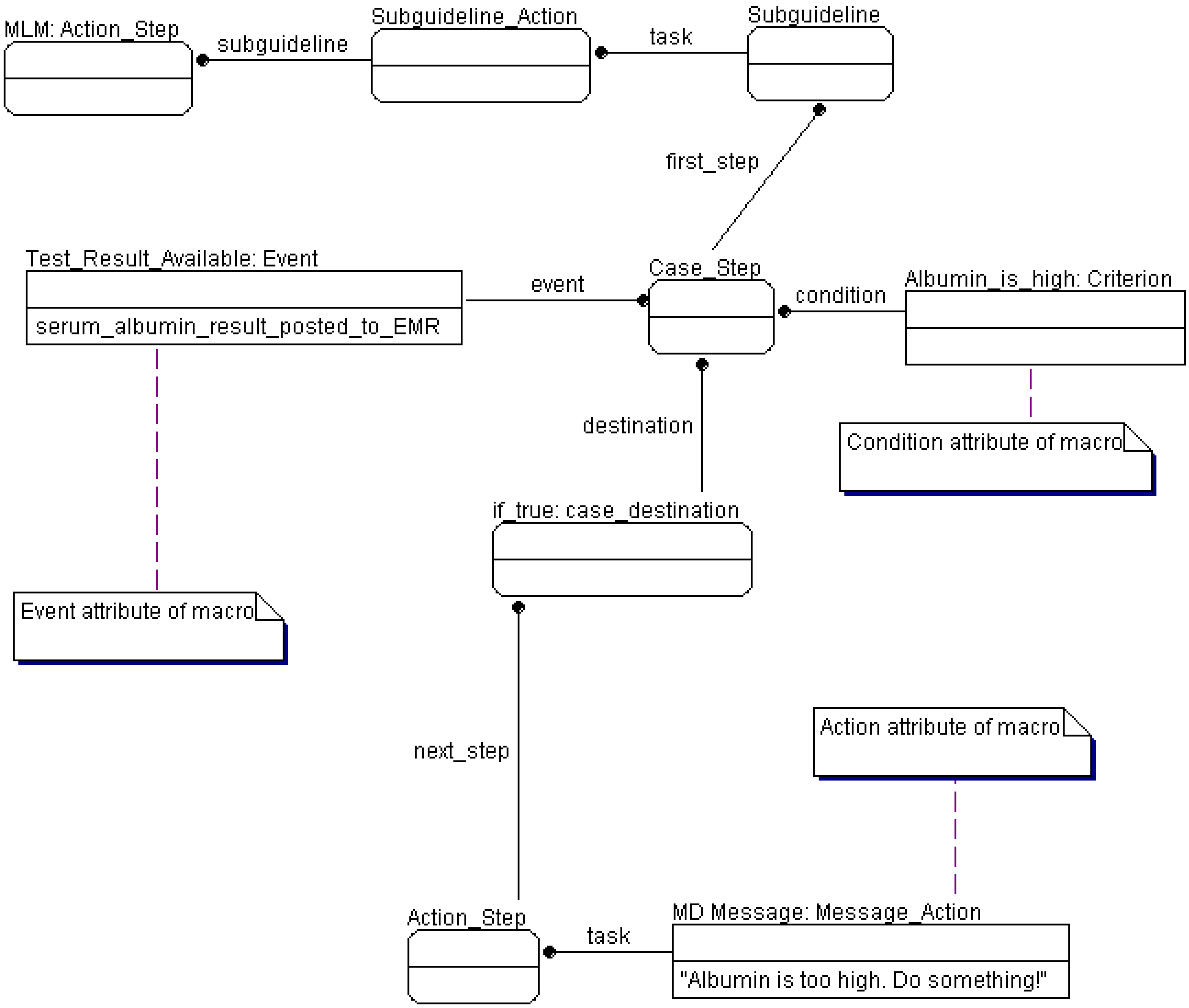
This is an example of a Macro ("MLM Macro") that illustrates how an Arden Syntax MLM can be described with a GLIF Macro. The Arden Macro object (a subclass of MacroStep) has 4 attributes, 3 of which map to MLM attributes, and one attribute which links in the next step. The three MLM attributes are

1. Evoke
2. Condition
3. Action

Note: This may also need a data slot.

Note:
true

Example:


MLM: Action_Step
subguideline
Subguideline_Action
task
Subguideline
first_step
Test_Result_Available: Event
serum_albumin_result_posted_to_EMR
event
Case_Step
condition
Albumin_is_high: Criterion
Condition attribute of macro
destination
if_true: case_destination
Event attribute of macro
next_step
Action attribute of macro
Action_Step
task
MD Message: Message_Action
"Albumin is too high. Do something!"

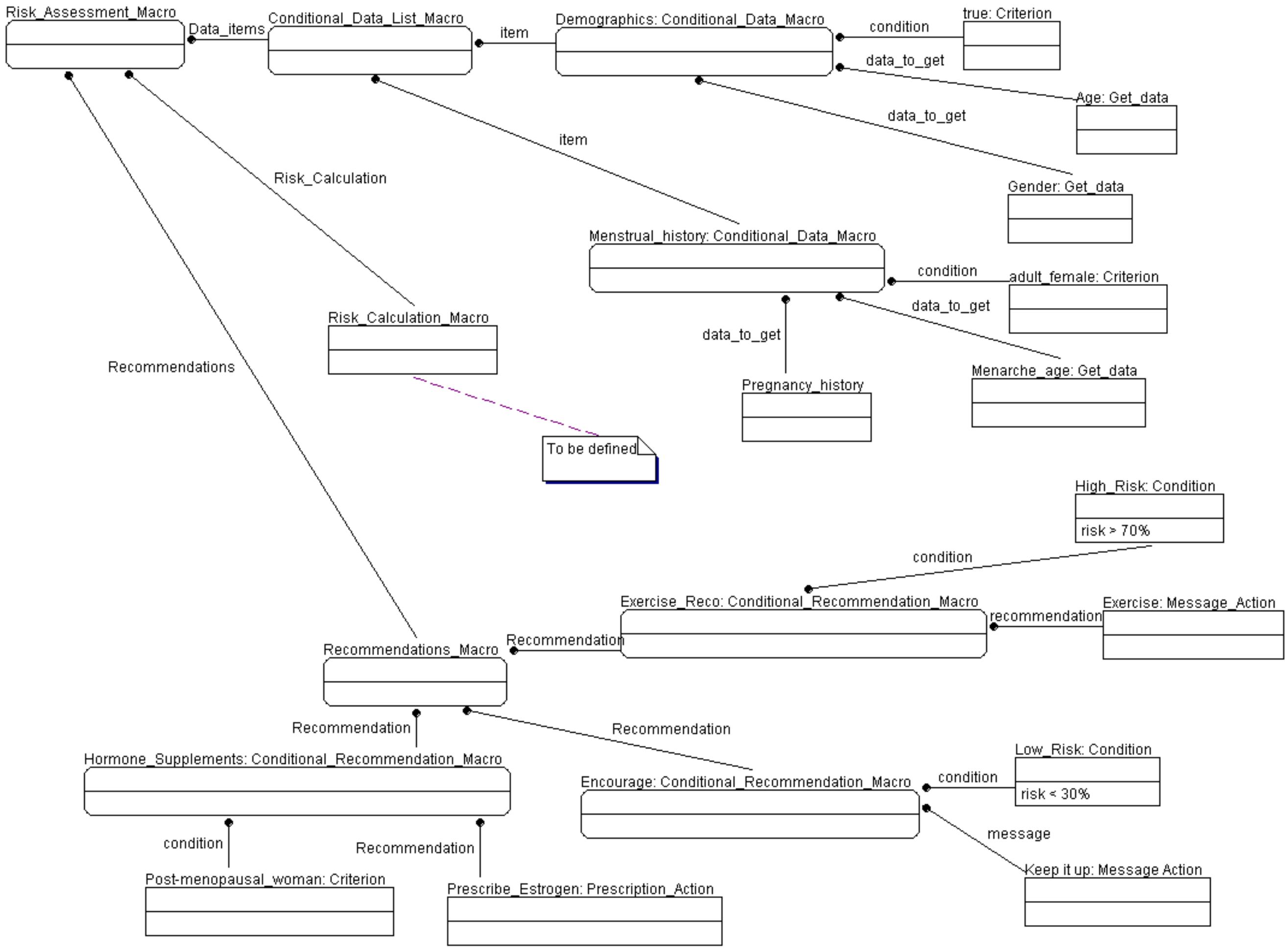
This MLM macro from MLM Macro 1 is expanded into a full GLIF object.

Note:
Event attribute of macro

Note:
Condition attribute of macro

Note:
Action attribute of macro

Example:

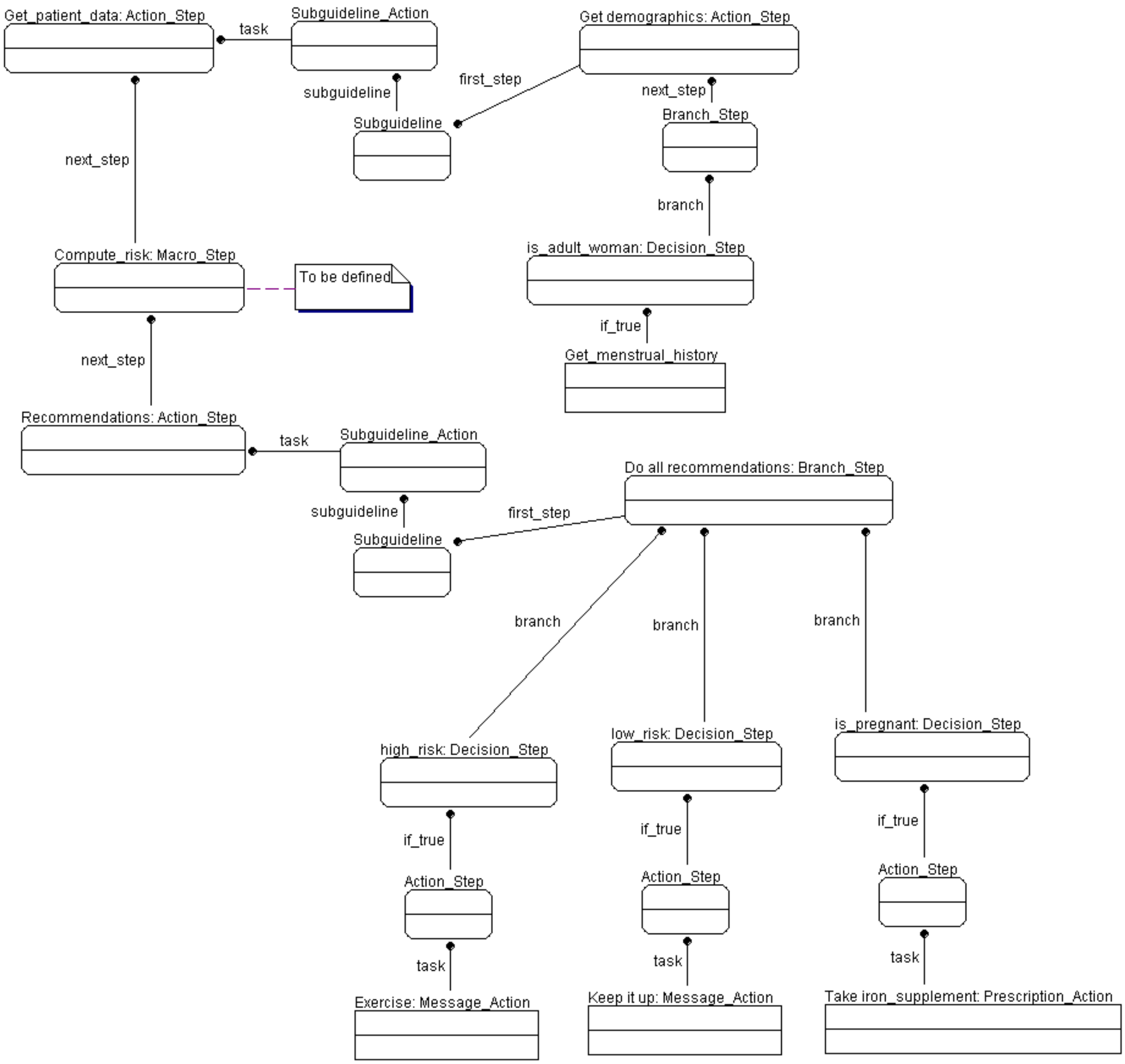


A generic risk assessment guideline contains three parts or "steps". Each of these steps can be modeled as a macro.

1. **Collecting patient data** - Data that is needed for calculating risk is collected in this step. The data may be obtained conditionally based on values of previously collected data. In this example, demographics data is obtained for all patients. Menstrual history is obtained only if the condition adult female is true.
   The data collection step is modeled using an ordered list of Get_Conditional_Data macros. This macro contains a condition and a list of patient data items that must be obtained.
2. **Computing risk** - The risk calculations are performed in this step. This macro has to be defined. It would contain definitions of variables that are to be created and the calculation of those variables in Assignment_Actions.
3. **Recommendations** - Recommendations based on computed risk and individual risk factors are provided in this step. Recommendations are provided only if an associated condition is true. The Exercise_recommendation is provided only to high risk persons.
   Thus, the Recommendations_Macro is structurally similiar to the Get_data_macro.

Note:
To be defined

Example:



This example expands the Risk Assessment Macro into its full GLIF implementation. That is, the macro has been expanded into more atomic GLIF steps.

Note:
To be defined