

## Package *Guideline\_Step\_Package.Decision\_Step\_Package*



### *Classes*

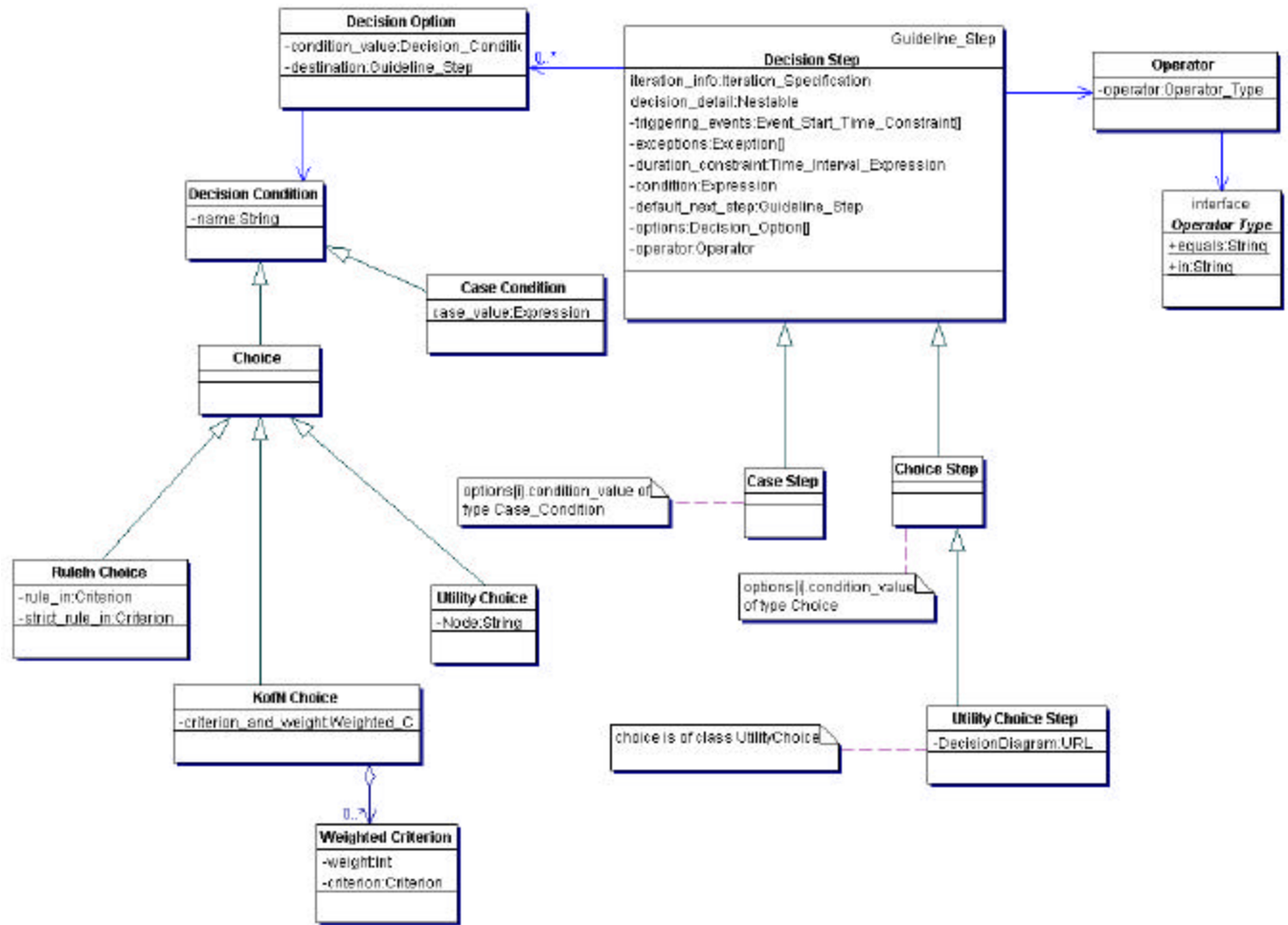
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Case\\_Condition](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Case\\_Step](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Choice](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Choice\\_Step](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Decision\\_Condition](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Decision\\_Option](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Decision\\_Step](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.KofN\\_Choice](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Operator](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.RuleIn\\_Choice](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Utility\\_Choice](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Utility\\_Choice\\_Step](#)  
**class** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Weighted\\_Criterion](#)



### *Interfaces*

**interface** [Guideline\\_Step\\_Package.Decision\\_Step\\_Package.Operator\\_Type](#)

## *Class Diagram*



**Note:**  
options[i].condition\_value of type Case\_Condition

**Note:**  
options[i].condition\_value of type Choice

**Note:**  
choice is of class UtilityChoice

## Interface Node Detail

### Interface *Guideline Step Package.Decision Step Package.Operator Type*

The operator "in" is used for intervals. For example, "x in (3,5), means  $x > 3$  and  $x < 5$ .

## Class Detail

### Class *Guideline Step Package.Decision Step Package.Case Condition*

Inherits from:

Guideline\_Step\_Package.Decision\_Step\_Package.Decision\_Condition

### Attributes

case\_value

## Attribute Detail



### case\_value

Data type: Expression

Multiplicity: 1

Description: the value of an expression to which the Case\_Step.case\_expression is compared

Level: B



### Class *Guideline\_Step\_Package.Decision\_Step\_Package.Case\_Step*

#### Inherits from:

Guideline\_Step\_Package.Decision\_Step\_Package.Decision\_Step

#### Description:

*Case steps* represent a deterministic decision between guideline steps. The different decision options are mutually exclusive. When a decision option's condition value matches the case step's condition, then control flows to the destination specified by that decision option.

#### IMPORTANT:

options.condition\_value must be of class CaseDestination.

#### Purpose:

#### Considerations:

1. The name of the case step will probably change when we complete the decision hierarchy.

#### Note:

options[i].condition\_value of type Case\_Condition



### Class *Guideline\_Step\_Package.Decision\_Step\_Package.Choice*

#### Inherits from:

Guideline\_Step\_Package.Decision\_Step\_Package.Decision\_Condition

#### Description:

*Choices* are used by *choice steps* to direct flow from one *guideline step* to another. A *choice* may be one of several kinds: *RuleInRuleOutChoice*, *KofNChoice*, and *UtilityChoice*. There may be other kinds of choices depending on how these choices are ranked.



### Class *Guideline\_Step\_Package.Decision\_Step\_Package.Choice\_Step*

#### Inherits from:

Guideline\_Step\_Package.Decision\_Step\_Package.Decision\_Step

#### Description:

*Choice steps* represent a non-deterministic decision between guideline steps. The different decision options are not necessarily mutually exclusive. Ranking the decision options depends on the class of choices. In a *ruleInRuleOutChoice*, for example, when a decision option's condition value's rule in and strict rule in matches the choice step's condition, then control may flow to the destination specified by that decision option. (1) If the criterion matches more than one of a set of values specified by the choices, a person will have to determine the next guideline step at run-time. Only one guideline step will be chosen to be the next step to be executed. (2) Each option contains a degree of preference that may be modeled differently for the different types of choices. (3) The degree of preference will determine how the choices will be ranked. This will assist the user in choosing among the different options.

#### IMPORTANT:

options.condition\_value must be of class Choice.

#### Purpose:

#### Considerations:

**Note:**  
options[i].condition\_value of type Choice

## Class *Guideline\_Step\_Package.Decision\_Step\_Package.Decision\_Condition*

### **Description:**

*Decision conditions* are classified as either case destinations or choices depending on whether the decision step that contains it is a case step or a choice step, respectively. It contains a name and a value that determines the next step for a case step or the rank of a choice for a choice step.

### **Attributes**

name

### **Attribute Detail**

#### **name**

Data type: String  
Multiplicity: 1  
Description: the name of a case destination or choice  
Level: A,B,C

## Class *Guideline\_Step\_Package.Decision\_Step\_Package.Decision\_Option*

### **Description:**

*Decision Options* are used by *Decision steps* to direct flow from one *guideline step* to another. Each *decision option* consists of a *Decision Condition* and a destination *guideline step*. The *condition* of a *decision step* is compared to the *condition value* of a *decision option*. If they are equal, then the control flows to the *guideline step* that is specified by the *decision option*.

### **Attributes**

condition\_value

destination

### **Attribute Detail**

#### **condition\_value**

Data type: Decision\_Condition  
Multiplicity: 1  
Description: the value of an expression to which the Decision\_Step.expression is compared  
Level: B

#### **destination**

Data type: Guideline\_Step  
Multiplicity: 1  
Description: the destination step that is taken when a Decision\_Step.expression equals the Decision\_Option.destination  
Level: A, B

## Class *Guideline\_Step\_Package.Decision\_Step\_Package.Decision\_Step*

### **Inherits from:**

Guideline\_Step\_Package.Guideline\_Step

### **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)

### Attributes

[condition](#)  
[decision\\_detail](#)  
[default\\_next\\_step](#)  
[duration\\_constraint](#)  
[exceptions](#)  
[iteration\\_info](#)  
[operator](#)  
[options](#)  
[triggering\\_events](#)

### Attribute Detail

#### **condition**

Data type: Expression  
Multiplicity: 1  
Description: the expression whose value is compared to the different decision options  
Level: B

#### **decision\_detail**

Data type: Nestable  
Multiplicity: 0:1  
Description: This optional attribute is used for specifying a nested condition. The execution of this nested item (a subguideline or macro) would precede the evaluation of the decision to be made by the step. The nested object would create or modify "variables" that are used in the decision criterion.  
Level: B

#### **default\_next\_step**

Data type: Guideline\_Step  
Multiplicity: 0:1  
Description: the step that follows when none of the choices or case destinations are feasible  
Level: B

#### **duration\_constraint**

Data type: Time\_Literal\_Expression  
Multiplicity: 0:1  
description: An interval whose start time is the shortest time that the step should take and whose end time is the maximum time that a step could take.  
Level: B

#### **exceptions**

Data type: Exception  
Multiplicity: 0:\*  
description: the set of exceptions that upon their occurrence a new next step is entered.  
Level: B

#### **iteration\_info**

Data type: Iteration\_Specification  
Multiplicity: 0:1  
Description: the iteration specification is compared  
Level: A, B and C

#### **operator**

Data type: Operator  
Multiplicity: 1

Description: a comparison operator used to compare the case expression to the different case values of the case step's destinations  
Level: B

### **options**

Data type: Decison\_Option  
Multiplicity: 1:\*  
Description: the list of conditions and their corresponding destination steps  
Level: B

### **triggering\_events**

Data type: Event\_Start\_Time\_Constraint  
Multiplicity: 0:\*  
Description: set of events that can trigger the step and their associated timing constraints. If several triggering events are defined for a step, then the step can occur whenever one of the events occur, provided that you are in the previous step.  
Level: B

## **Class [Guideline Step Package.Decision Step Package.KofN Choice](#)**

### **Inherits from:**

Guideline\_Step\_Package.Decision\_Step\_Package.Choice

### **Description:**

*KofNChoices* contain an array of criteria, each associated with a weight. The weighted criteria for each of the options will determine how an option will be ranked among the choices presented to a user at run-time. The sum of the weights for each criterion in a choice has to equal 1.0. The higher the value of a choice (from 0 to 1), the higher its rank.

### **Attributes**

[criterion and weight](#)

### **Attribute Detail**

### **criterion\_and\_weight**

Data type: Weighted\_Criterion  
Multiplicity: 1:\*  
Description: the list of criteria and weights that establish a numerical value for this choice  
Level: B

## **Class [Guideline Step Package.Decision Step Package.Operator](#)**

The operator is used to compare the case expression in a decision step to the actual case value in a case step.

### **Attributes**

[operator](#)

### **Attribute Detail**

### **operator**

Data type: Operator\_Type  
Multiplicity: 1  
Description: used to compare a case expression with a case value  
Level: A,B,C

## **Class [Guideline Step Package.Decision Step Package.RuleIn Choice](#)**

### **Inherits from:**

Guideline\_Step\_Package.Decision\_Step\_Package.Choice

### **Considerations:**

1) It is possible to add two more attributes: rule\_out and strict\_rule\_out, but we can always use the "not" operator (e.g., rule\_in: not A or rule\_out: A), and so the rule\_out and strict rule\_out are not required.

Example: Strict-rule-in criterion : Penicillin allergy=false vs. Strict-rule-out criterion: Penicillin allergy= true for giving amoxicillin prophylaxis for bacterial endocarditis.

2) If the criterion language will distinguish between criteria and strict criteria, then we can use just the rule\_in attribute. The question is whether this distinction is required elsewhere, or just in making choices.

### Attributes

[rule\\_in](#)  
[strict\\_rule\\_in](#)

### Attribute Detail

#### rule\_in

Data type: Criterion  
Multiplicity: 1  
Description: the value may either be true or false and plays a role in ranking this choice  
Level: B

#### strict\_rule\_in

Data type: Criterion  
Multiplicity: 0:1  
Description: the value may either be true or false and plays a role in ranking this choice as best choice if true or worst choice if false  
Level: B

### Class [Guideline Step Package.Decision Step Package.Utility Choice](#)

#### Inherits from:

Guideline\_Step\_Package.Decision\_Step\_Package.Choice

#### Description:

*UtilityChoices* contain the node that will determine its expected value or expected utility at run time. This expected utility will determine its rank among the list of choices.

### Attributes

[Node](#)

### Attribute Detail

#### Node

Data type: String  
Multiplicity: 1  
Description: the expected value of a choice based on a decision analysis model  
Level: B

### Class [Guideline Step Package.Decision Step Package.Utility Choice Step](#)

#### Inherits from:

Guideline\_Step\_Package.Decision\_Step\_Package.Choice\_Step

#### Description:

*Utility Choice steps* represent a choice step that uses the Utility theory in deciding among several options. It contains a pointer to the actual algorithm used to evaluate the choices. This may either be a decision analysis tree or an influence diagram.

#### IMPORTANT:

**options.condition\_value must be of class UtilityChoice.**

#### Note:

choice is of class UtilityChoice

### Attributes

[DecisionDiagram](#)

### Attribute Detail

#### DecisionDiagram

Data type: URL  
Multiplicity: 1  
Description: the location of the utility diagram/decision analysis model  
Level: A,B,C

### **Class** [Guideline Step Package.Decision Step Package.Weighted Criterion](#)

The weighted criterion contains the case value or the expression to be compared against the actual case expression in a decision step. This is assigned a weight that together with the actual numerical case value determines the rank of a particular choice.

#### **Attributes**

[criterion](#)  
[weight](#)

#### **Attribute Detail**



##### **criterion**

Data type: CriterionL  
Multiplicity: 1  
Description: the criterion may evaluate to true or false  
Level: B



##### **weight**

Data type: int  
Multiplicity: 1  
Description: the weight given to a criterion  
Level: A,B,C

#### **Interface Detail**



### **Interface** [Guideline Step Package.Decision Step Package.Operator Type](#)

The operator "in" is used for intervals. For example, "x in (3,5), means  $x > 3$  and  $x < 5$ .

#### **Attributes**

[equals](#)  
[in](#)

#### **Attribute Detail**



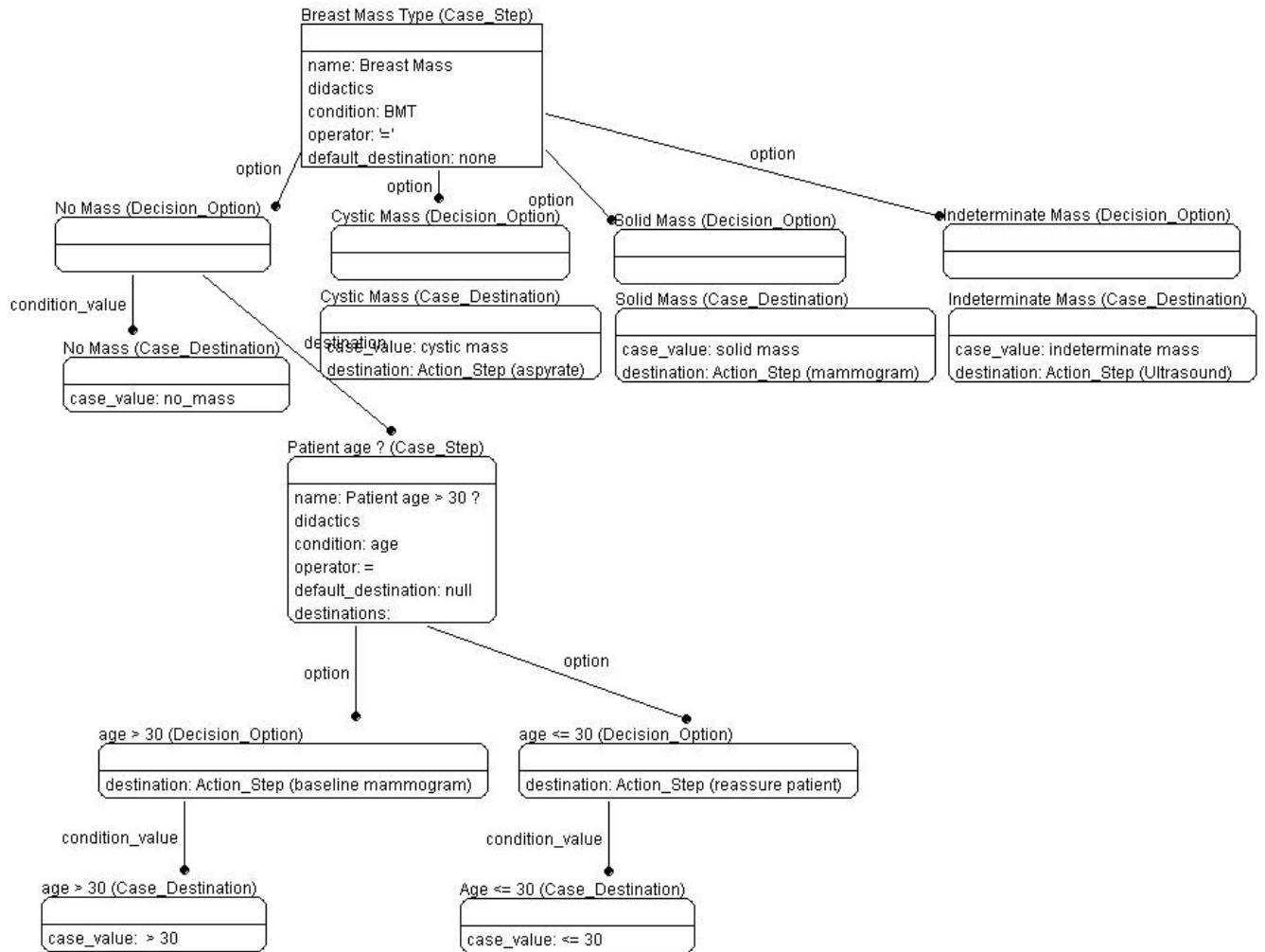
##### **equals**



##### **in**

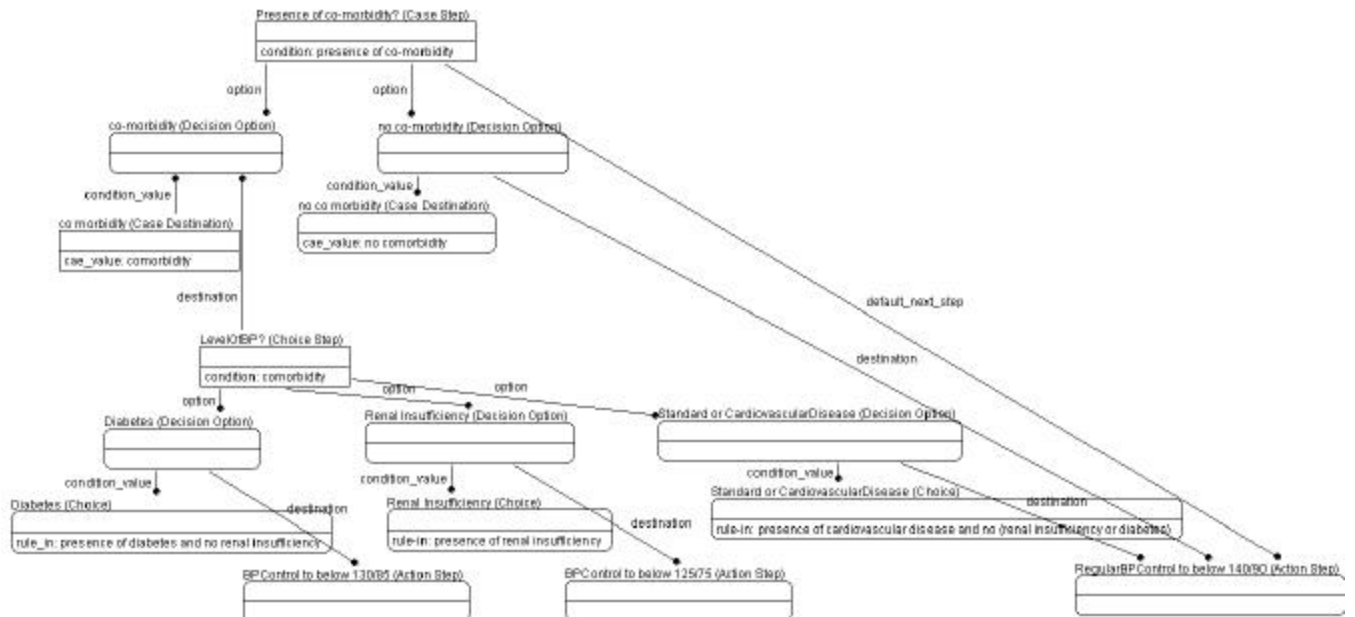
**Example:**



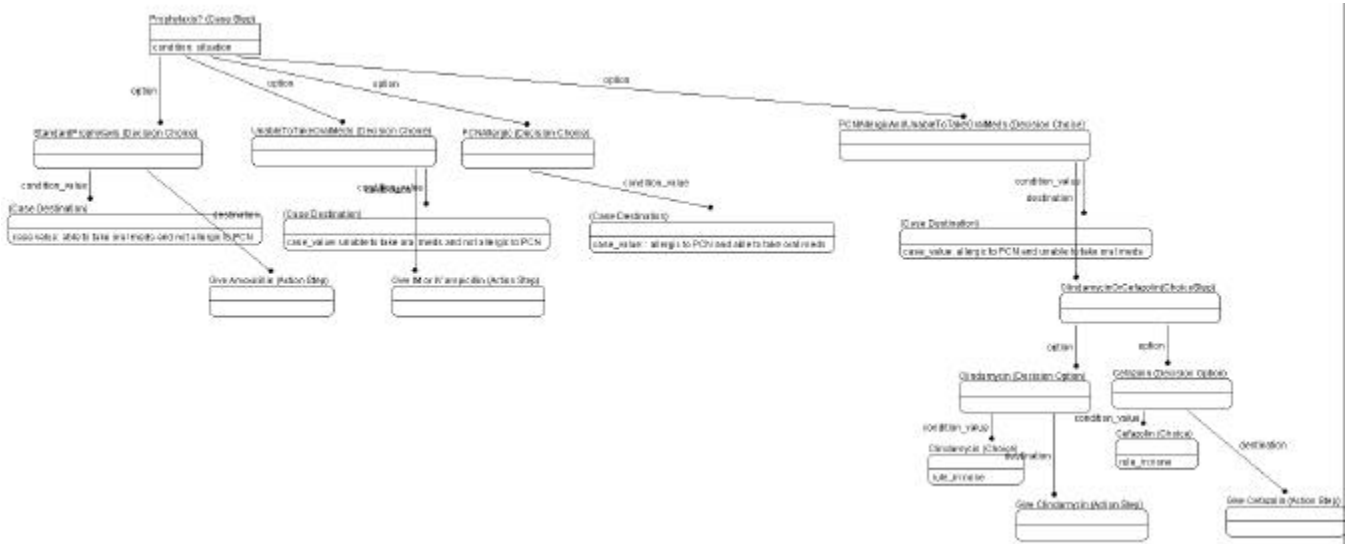


In this example, the case step, named "Breast Mass Type" is shown on the top. There are 4 possible destinations, depending on the value of the criterion BMT. If BMP is equal to "no mass" then the destination is the case step "patient age > 30?"; If BMP is equal to "solid mass" then the destination is the action step "mammogram"; If BMP is equal to "cystic mass" then the destination is the action step "aspirate cyst "; If BMP is equal to "indeterminate mass" then the destination is the action step "Ultrasound"

**Example:**



Example:



This is based on the American Heart Association's prevention of Bacterial Endocarditis Guideline. (Prophylactic Regimens for Dental, Oral, Respiratory Tract, or Esophageal Procedures).

The case step in example 2 is a soft case step and may actually be modeled as a choice step. The criteria in a case step are all absolute. One may be able to take oral meds, but still opt to take Ampicillin IV.