GLIF

Mon Feb 14 16:18:19 PST 2000

Package *Views_Package*

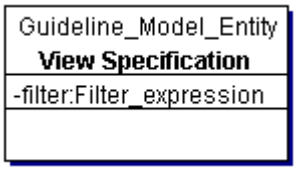
Subpackages

package Views_Package

Classes

class Views_Package.View_Specification

Class Diagram



Class Detail

Class *Views_Package.View_Specification*

Inherits from:

Guideline_Model_Entity

Description:

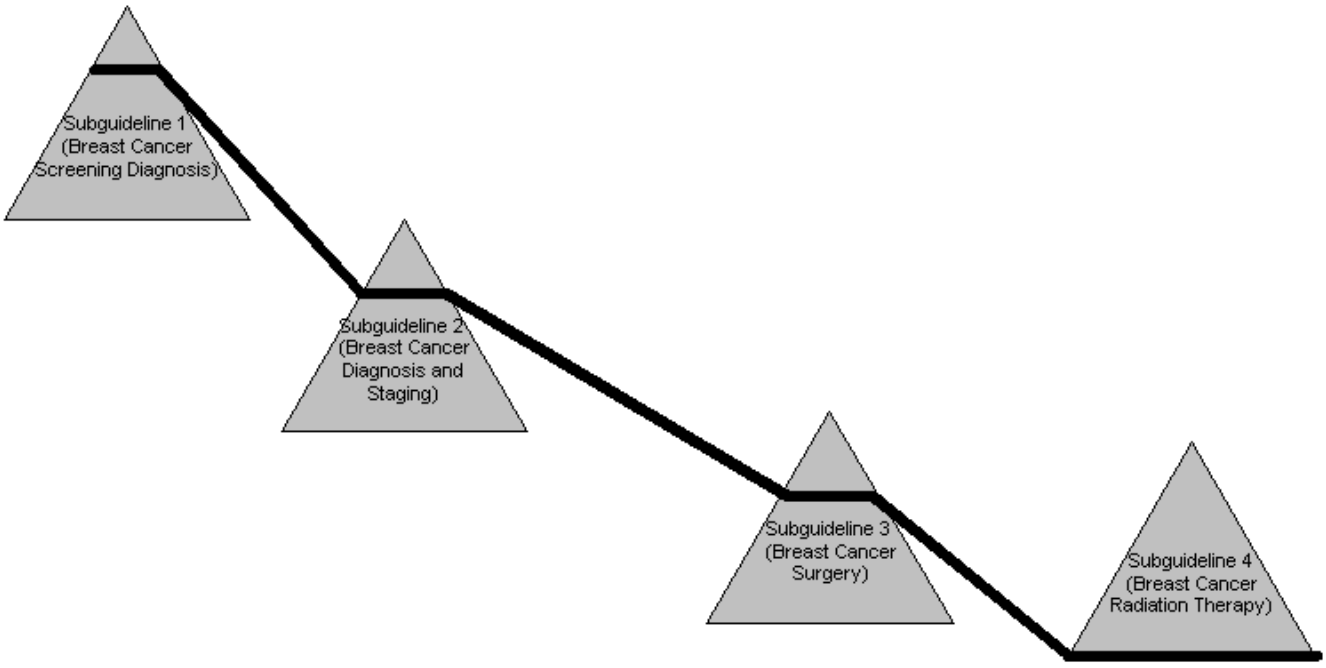
Note that any user should be able to see any entity in arbitrary detail, but only the specified user will see this level of detail by default .

Purpose:

Views are a way of managing the complexity of guidelines.

If we accept the thesis that one purpose of guidelines is to reduce practice variability in favor of superior patterns, it seems likely that guidelines with greater scope will eventually appear.

Views describe the amount of detail (i.e. level of nesting) displayed by default for each of the subguidelines.



If a guideline consists of Subguidelines, each of these subguidelines may be visualized as a triangle, with one step at the highest level and multiple steps at the lowest level. That is, the width of the triangle is proportional to the number of steps at that level of nesting. Top of the triangle = less detail = small number of steps. Bottom of the triangle = more detail = large number of steps.

A given filter (e.g. MD_Radiation_Oncologist) will define the default level of nesting/zooming for each of the subguidelines. It will be up to the guideline author to define the subguidelines in an appropriate way (e.g. to avoid too many steps per screen for a given viewer) and to define the level of nesting required for each given subguideline. In the above example, suppose that a breast cancer guideline has four subguidelines as above. A Radiation Oncologist looking at the guideline may see (by default) relatively little detail about screening, diagnosis and surgery. He will see a great deal of detail regarding radiation therapy, however. A surgeon looking at the same guideline may see little detail re: screening/diagnosis, a lot on surgery and little re: radiation therapy.

The status quo of specialty bodies publishing guidelines may change as multi-specialty organizations publish multi-specialty documents. Guidelines may become quite complex.

Much of medicine is multi-disciplinary in nature. The distinction between specialties is artificial. For example, the distinction between cardiology and nursing is for the convenience of practitioners. The patient suffering a myocardial infarction (heart attack) is likely to require care from both a cardiologist and a nurse. The information needs of the cardiologist, however, are very different from those of the nurse. The purpose of default views in GLIF should be to reveal to the cardiologist only the relevant portions of the myocardial infarction guideline, which may be different from that shown to the nurse.

Consideration:

Views are default filters through which we interact with the guideline. By definition, views do not change guideline logic (e.g. if an RN should do something different from an MD, this should be represented in the guideline logic, not in the view). Although we anticipate that the most common use of views will be user and/or location, there may be other relevant filters (e.g. situation such as routine vs. disaster). The view class is a guideline entity. Alternatively, the view could have been modeled as an enumerated type attribute. The main purpose of this class is to allow differential display in the simplest possible way.

The view specification was chosen to be at the level of guideline entities and not at the attribute level. We may later choose to make **attributes** (and not entire guideline entities) visible or invisible to some users.

Issues yet to be resolved:

1. Semantics of no default_user specified (e.g. confidential vs. visible to developer only).

2. Issues related to nesting?

Attributes

filter

Attribute Detail

filter

Data type: Filter_expression
Multiplicity: 0:1
Description: Views are a filter applied to the guideline, there may be multiple relevant filters, e.g. viewer (MD vs. RN), location (office vs. hospital), situation (disaster vs. routine), etc.

BNF:

term: filter_type = *domain_ontology_filter_instance*

filter_expression: term | *expression binary_operator expression* | *unary_operator expression* | **(***expression***)**

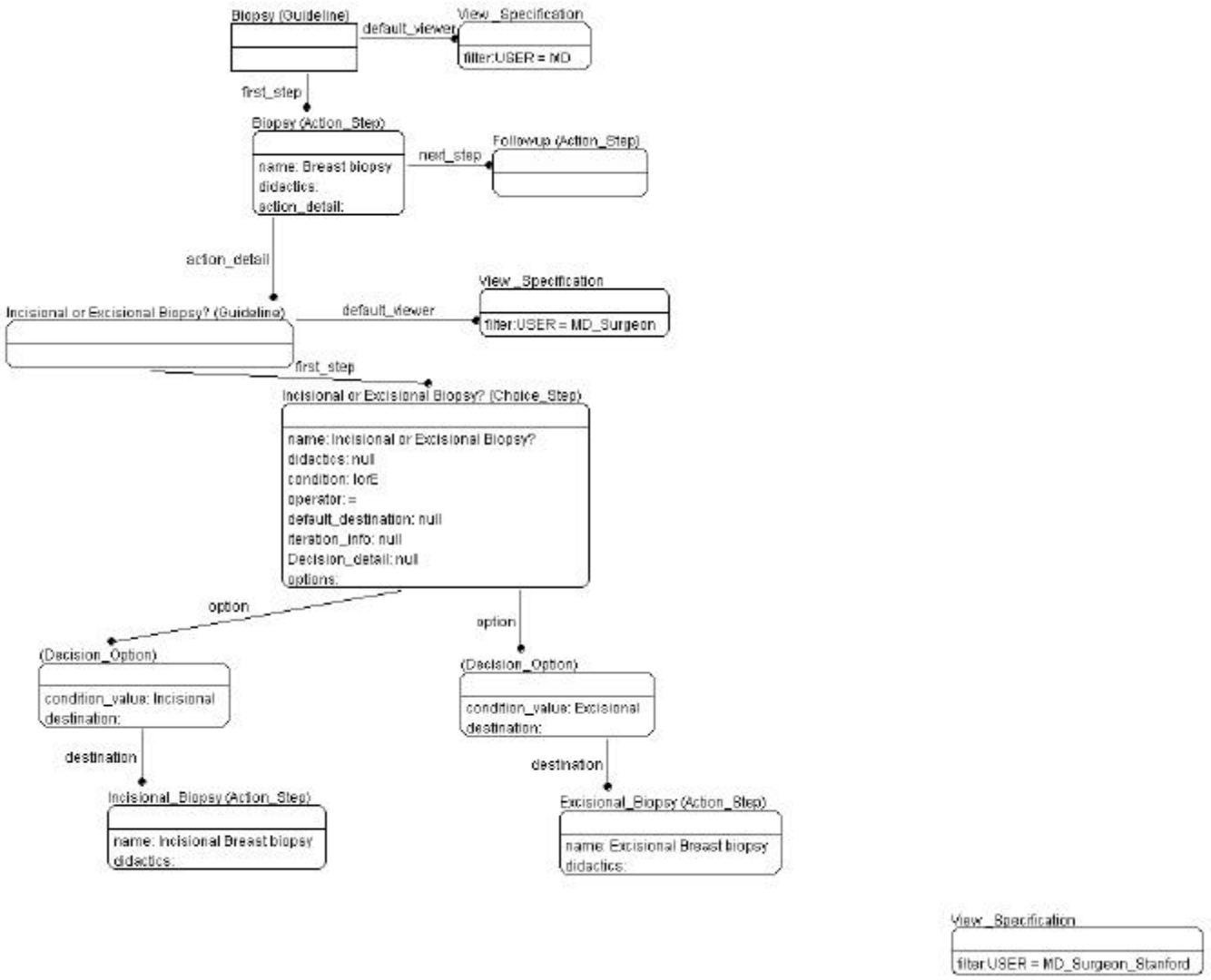
binary_operator: **OR** | **AND**
unary_operator: **NOT**

filter_type: **USER** | **LOCATION**

domain_ontology_filter_instance: **MD** | **RN** | ...

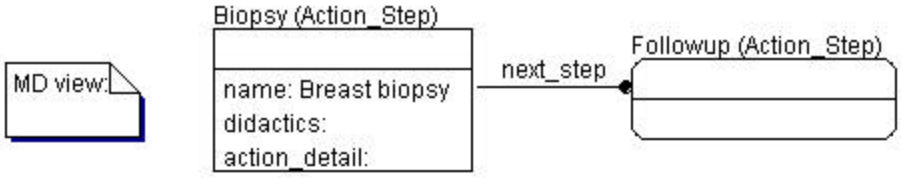
Level: A, B and C

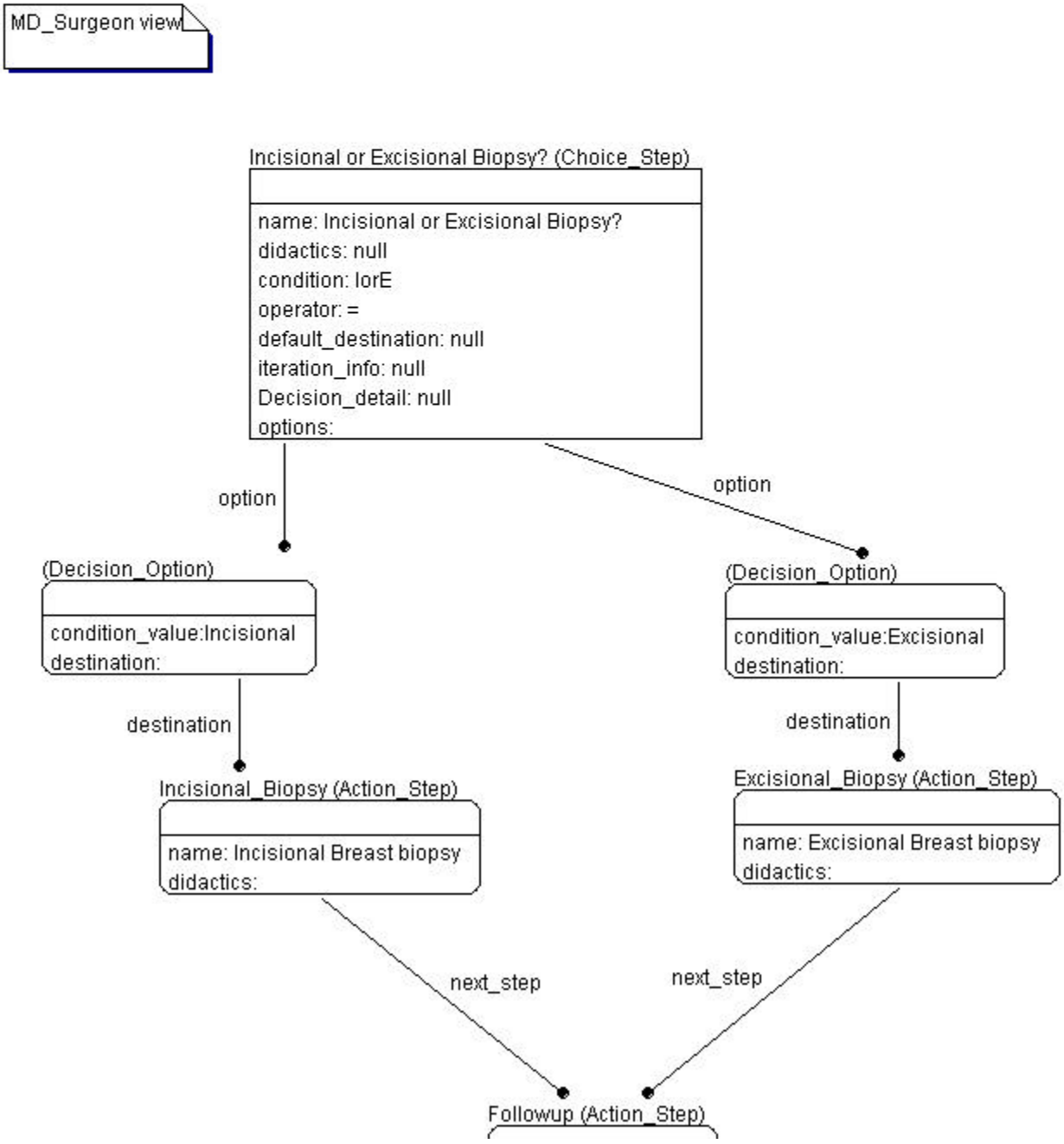
Example:



A guideline might call for a breast biopsy. Lets say that all MDs want to see that a breast biopsy is called for, however, surgeons want to know what kind of biopsy is needed, incisional or excisional. .

Example:


MD view:
Biopsy (Action_Step)
name: Breast biopsy
didactics:
action_detail:
next_step
Followup (Action_Step)


MD_Surgeon view
Incisional or Excisional Biopsy? (Choice_Step)
name: Incisional or Excisional Biopsy?
didactics: null
condition: IorE
operator: =
default_destination: null
iteration_info: null
Decision_detail: null
options:
option
option
(Decision_Option)
condition_value:Incisional
destination:
(Decision_Option)
condition_value:Excisional
destination:
destination
destination
Incisional_Biopsy (Action_Step)
name: Incisional Breast biopsy
didactics:
Excisional_Biopsy (Action_Step)
name: Excisional Breast biopsy
didactics:
next_step
next_step
Followup (Action_Step)

This example shows how nesting deals with views. If the viewer is an MD he sees the top-level view of the action step Biopsy. He can zoom into the action-detail subguideline, to see that incisional or excisional biopsies can be performed.

An MD_Surgeon will directly see the zoomed-in view of biopsy directly, showing the decision that is made between incisional and excisional biopsy.

Note:
MD view:

Note:
MD_Surgeon view