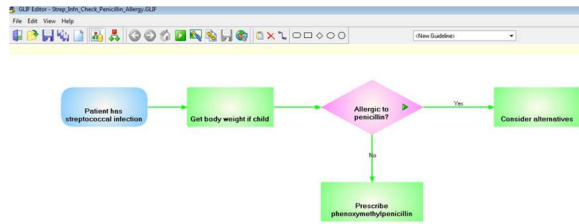


GLIF and GELLO worked example - allergies

Now we will do an example for checking whether a patient is allergic to penicillin.

Set up a simple GLIF file such as the following, remember to click on the the **Allow Modifications** button to enable editing:



Give it a name such as *Strep_Infn_Check_Penicillin_Allergy.GLIF*

Load the patient data file test.xml using the GLIF editor top menu as in the previous tutorial.

Go to the 'Yes' arrow outflowing from the "Allergic to penicillin?" Decision step, and open the GELLO editor for this decision execution.

Load up the vmr as before, but this time we are interested in patient allergies , so make the code to be:

```
Context GLIF_VMR::GLIFDecisionNode  
  
vmr.allergies
```

Name	Class	Data
allergyCode	CD	6369005
allergyCode	CD	2.16.840.1.113883.6.96

Ok we can see two allergies. Unfold on the allergenCode attribute of the second of the two listed allergies in Model Explorer. Note the mandatory three properties of a CD datatype again - the code, codeSystem and codeSystemName. In this case they are '6369005', '2.16.840.1.113883.6.96' and 'SNOMED-CT'.

Set up the local variable for a parent SNOMED CT concept we want to query on. (ie this code will need to cover *all* penicillins in SNOMED CT). Note that in SNOMED CT , medications can be products or substances. Use an online SNOMED CT browser such as that available at <http://browser.ihtsdo.org/>

Please note to take care to comply with license restrictions of SNOMED International

Concept	SNOMED CT	codeSystemName
Penicillin	P01AD01	SNOMED-CT

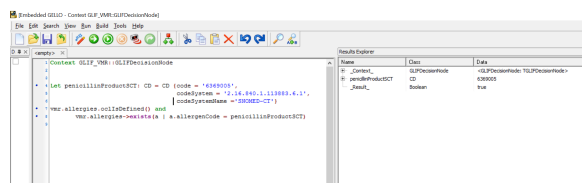
So the code we currently have for the patient is already a high level concept in the Product hierarchy in SNOMED CT. A parent concept for this would be 90614001 | beta-Lactam antibiotic (product) |, but this will subsume cephalosporins as well (while there is a potential cross allergy to this class, many patients can tolerate them). So we will use the code '6369005', and it just happens to be the same as the code in the patient record in this particular case.

Let's just use an 'equals' operator for now in the last line:

```
Let penicillinProductSCT: CD = CD {code = '6369005',
                                   codeSystem = '2.16.840.1.113883.6.1',
                                   codeSystemName = 'SNOMED-CT'}

vmr.allergies.ocIsDefined() and
vmr.allergies->exists(a | a.allergenCode = penicillinProductSCT)
```

Here's the result:



The last line can become

```
vmr.allergies->select(a|a.allergenCode.implies(penicillinProductSCT).
asBoolean())>size()>0
```

or

```
vmr.allergies->exists(a|a.allergenCode.implies(penicillinProductSCT).
asBoolean())
```

and this is different for a couple of reasons. Firstly and importantly we are now using the *implies* method which looks for parent-child subsumption in the SNOMED CT ontology (in this case defaulting to "=" as the codes are the same); and secondly with respect to syntax we are using a select or exists operator on the collection of allergies. The syntax can look a bit daunting the first time, but this is an example of some GELLO code you can cut and paste and then modify for new uses. Also the implies method may not work in your system depending on whether we have hooked our SNOMED CT server up for you or not.

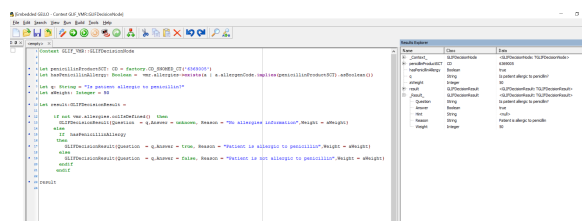
The setting up of the local CD variable can also be changed and shortened if you wish:

```
Let penicillinProductSCT: CD = factory.CD_SNOMED_CT('6369005')
```

In any case we need to make this a GLIF Decision, so modify the last line by saying

```
Let hasPenicillinAllergy: Boolean =
```

ahead of what is there and then insert a GLIFDecisionResult-typed result as in this screenshot:



So now we save all that and then copy this gello into the "no" arrow and modify the GLIF Decision result code so we get a corresponding boolean "no" in that executed decision arrow. This section for the "no" arrow becomes:

```
Let q: String = "Is patient allergic to penicillin?"
Let aWeight: Integer = 50

Let result:GLIFDecisionResult =

    if not vmr.allergies.ocIsDefined() then
        GLIFDecisionResult{Question = q,Answer = unknown, Reason = "No
allergies information",Weight = aWeight}
    else
        If not hasPenicillinAllergy
        then
            GLIFDecisionResult{Question = q,Answer = true, Reason = "Patient is
not allergic to penicillin",Weight = aWeight}
        else
            GLIFDecisionResult{Question = q,Answer = false, Reason = "Patient
is allergic to penicillin",Weight = aWeight}
        endif
    endif

result
```