Using patient states as intermediate state objects (includes Sync node GELLO)

Overview

The last tutorial showed how archetypes can hold data for persistence and reuse with some GLIF. While this is effective, an alternative approach is to use patient states as intermediate state objects. In other words the patient state can hold a value which is then accessible to GELLO, within the GLIF application. Currently we are allowing Boolean True False and Unknown but it is a work in progress. This approach avoids encroaching on GELLO immutability upon the EHR (GELLO is a functional language) , which could technically be an issue when using archetypes (archetypes can be thought of as being part of the information model; and a tenet of GELLO is that it is side effect free). This tutorial will show a couple of examples of this use including GELLO for patient states. It finishes with showing GELLO for Sync nodes as alluded to in an earlier tutorial.

Patient states as intermediate state objects

This example will demonstrate how we can store a boolean value of True or False in a patient state for use later on in the GLIF flowchart.

First we will need some test data, in this case we will use a xml format. Save the following at *GLIF_TutorialSix.xml*

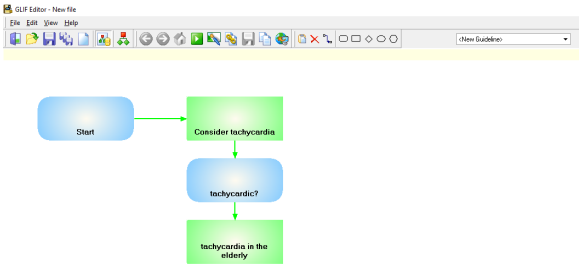
```
<?xml version="1.0" encoding="UTF-8"?>
<singlePatient xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:noNamespaceSchemaLocation="./iso-21090-datatypes.xsd">
    <patient>
        <patientID
            root="1.2.36.174030967"
            extension="1234567892"
            reliability="VRF"
            scope="OBJ"/>
        <patientName>
            <part type="GIV" value="Alice" />
            <part type="GIV" value="A." />
            <part type="FAM" value="Someone" />
        </patientName>
        <dob value="19350621"/>
        <gender code="248152002"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "Female" />
        </gender>
        <ssn value="987-65-4320"/>
        <address use="WP">
            <part type="AL" value="1050 W Wishard Blvd" />
            <part type="CTY" value="Indianapolis" />
            <part type="STA" value="IN" />
            <part type="ZIP" value="46240" />
        </address>
    </patient>
    <allergies>
        <allergenType code="373873005"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "pharmaceutical / biologic product" />
        </allergenType>
        <allergenCode code="111088007"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "latex (product)" />
        </allergenCode>
        <allergySeverity code="24484000"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "severe" />
        </allergySeverity>
        <allergyReaction code="39579001"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "anaphylaxis" />
        </allergyReaction>
        <identificationDate value="1980"/>
    </allergies>
```

```
    <allergies>
        <allergenType code="373873005"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "pharmaceutical / biologic product" />
        </allergenType>
        <allergenCode code="27658006"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "Amoxicillin" />
        </allergenCode>
        <allergySeverity code="24484000"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "severe" />
        </allergySeverity>
        <allergyReaction code="39579001"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "anaphylaxis" />
        </allergyReaction>
        <identificationDate value="1980"/>
    </allergies>
    <historyOfPastIllness>
        <pastIllness code="302914006"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "Barrett's esophagus" />
        </pastIllness>
        <temporalContext code="410587003"
            codeSystem="2.16.840.1.113883.6.96"
            codeSystemName="SNOMED-CT">
            <displayName value = "Past - specified" />
        </temporalContext>
        <illnessDates>
            <!--IVL_TS -->
            <low value="2000" />
        </illnessDates>
        <notesOnIllness></notesOnIllness>
    </historyOfPastIllness>
    <medications>
        <currentMedications>
            <!--1-->
            <medicationCode code="317334001"
                codeSystem="2.16.840.1.113883.6.96"
                codeSystemName="SNOMED-CT">
                <displayName value = "Esomeprazole 40mg tablet" />
                    <translation code="606731"
                        codeSystem="2.16.840.1.113883.6.88"
                        codeSystemName="RxNorm">
                        <displayName value = "Esomeprazole 40 MG Delayed
Release Oral Capsule [Nexium]" />
                    </translation>
            </medicationCode>
            <dosingSig code="229797004"
                codeSystem="2.16.840.1.113883.6.96"
                codeSystemName="SNOMED-CT">
                <displayName value = "Once daily" />
            </dosingSig>
            <requestedGiveRate></requestedGiveRate><!--PQ eg 100mls an hour-->
            <medicationStartDate value="20160128" />
            <medicationStopdate> </medicationStopdate>
            <totalDailyDose> </totalDailyDose>
            <reasonForCessation> </reasonForCessation>
            <indication code="302914006"
                codeSystem="2.16.840.1.113883.6.96"
                codeSystemName="SNOMED-CT">
                <displayName value = "Barrett's esophagus" />
            </indication>
            <replacedBy> </replacedBy>
        </currentMedications>
    </medications>
    <vitals>
        <pulseRate>
            <observationCode code="364075005"
                codeSystem="2.16.840.1.113883.6.96"
                codeSystemName="SNOMED-CT">
                <displayName value = "pulse rate" />
            </observationCode>
            <dateTime value = "20160618" />
            <value xsi:type = "PQ" value = "164" unit = "bpm" />
        </pulseRate>
    </vitals>
</singlePatient>
```

Note we have a date of birth and a pulse rate available.

Now open the GLIF Editor and go **File**, **New** if necessary to clear the workspace.Make sure you save anything you had there before that you want to use again (or you can open a new instance of the GLIFEditor - ie have two running at once - this can be handy when looking to cut and past GELLO code for example).

Put in an initial patient state linked to an action then link that to another patient state. Put an action linked to the second patient state. Name them 'Start', 'Consider tachycardia', 'tachycardic?' and 'tachycardia in the elderly'. Make the initial patient state the first step.



You can add some didactics to the action boxes, something like 'Tachycardia can represent significant illness' for the first, and 'Tachycardia in the elderly be the result of uncontrolled atrial fibrillation' for the second.

Ok, let's edit the 'tachycardic?' patient state. When you right click and go **Edit properties** notice the **Edit Gello** button in the bottom left of the window. Click on this.

Patient state GELLO is somewhat simpler in that we simply need to return a Boolean value rather than a GLIFDecisionResult class instance.

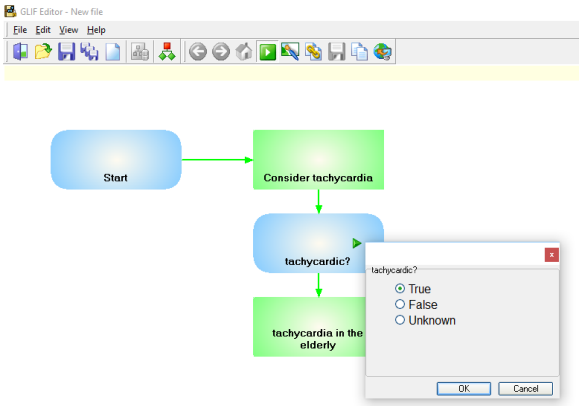
Here is the code to insert:

```
   Context GLIF_VMR::GLIFDecisionNode

   Let heartRate_SCT:CD = CD{code = '364075005',
       codeSystem = '2.16.840.1.113883.6.96',
       codeSystemName ='SNOMED-CT'
       }

   Let mostRecentPulseRate:PQ = If vmr.vitals.pulseRate.oclIsDefined() then
       vmr.vitals.pulseRate->select(observationCode=heartRate_SCT)->sortedBy
(dateTime)->last().value.oclAsType(PQ) else
         null
       endif

   Let has_Tachycardia: Boolean =
       if mostRecentPulseRate > factory.PQ(120, 'bpm') then true

           else false
       endif

   has_Tachycardia
```

Save that and save the GLIF file. Let's test what we have so far.

Go **File**, **Load xml Test Data** and choose *GLIF_TutorialSix.xml*

Click on the Run button. What happens? Yes the patient state gets a little green arrow in it and the link arrow from it to the next action step should remain green (ie it is running). Click on the green arrow and you can see the value being persisted for the patient state as a result of the GELLO that has been executed.

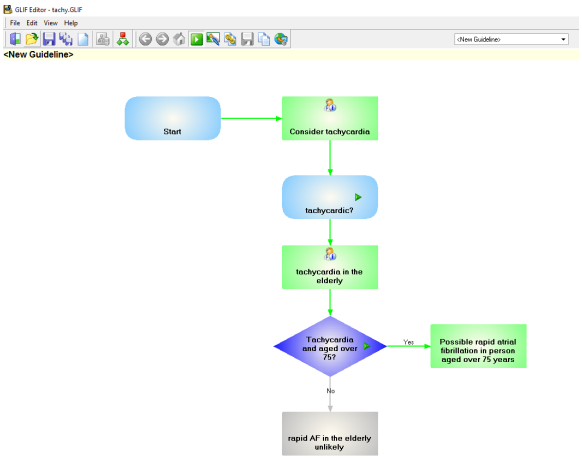


Now we will re-use that intermediate persisted state in a Decision. Add a decision node and call it 'Tachycardia and aged over 75?' Flow the decision out to two action nodes.

In the Yes link, use this code:

```
   Context GLIF_VMR::GLIFDecisionNode

   Let q: String = "Is patient tachycardic and aged over 75yrs?"
   Let aWeight: Integer = 50

   Let isTachycardic: Boolean = GLIFNodes.patient_state_by_name
('tachycardic?')
   Let age: PQ = factory.Now() - vmr.patient.dob
   Let ageInYears: PQ = age.convert('yr')

   Let result:GLIFDecisionResult =

     if not vmr.vitals.pulseRate.oclIsDefined() and not vmr.patient.dob.
oclIsDefined() then
       GLIFDecisionResult{Question = q,Answer = unknown, Reason = "No pulse
rate or age information",Weight = aWeight}
        else
          If isTachycardic and ageInYears >= factory.PQ(75, 'yr')
            then
            GLIFDecisionResult{Question = q,Answer = true, Reason = "Patient
is tachycardic and aged over 75 years",Weight = aWeight}
          else
           GLIFDecisionResult{Question = q,Answer = false, Reason = "Patient
is not tachycardic and aged over 75 years",Weight = aWeight}
        endif
     endif

    result
```

I will leave it you to put the corresponding code for the 'No' link. Line 6 in the editor view is the salient one here. The method 'patient_state_by_name' passes a parameter which is the name of the relevant patient state. GLIFNodes is a class which is in GLIFDecisionNode being in turn a class in the GLIF_VMR package. You can look at this in the Model Explorer. Here is the finished GLIF:

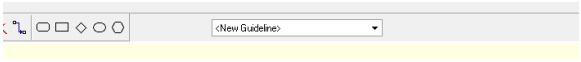


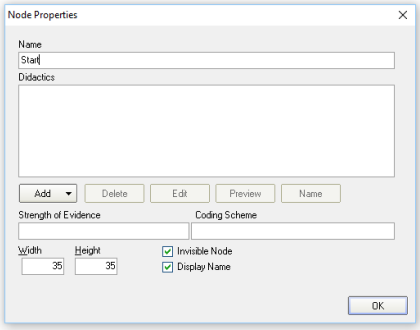
So this is showing how we can use a patient state as an intermediate state object.

Patient states as preconditions

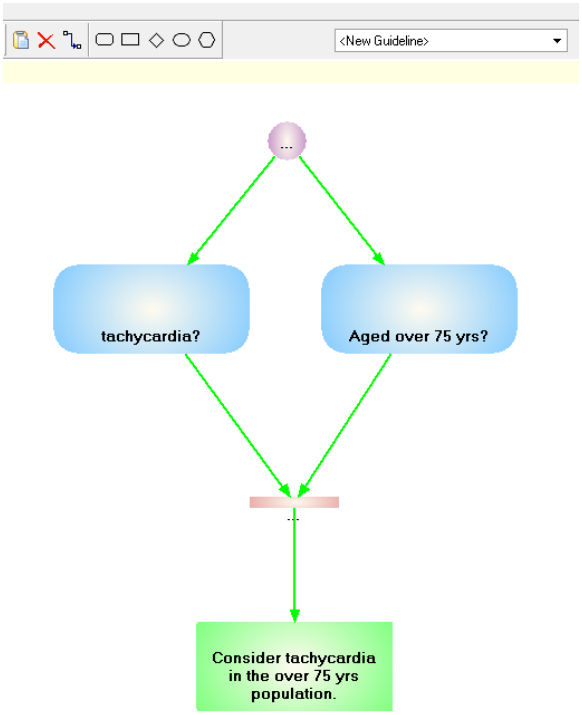
Another way to design things is to allow for some patient states as preconditions to then come together to produce an Action (a little bit like petri nets). Lets rearrange the GLIF we have done to demonstrate this.

Set up a new GLIF file and click on the **Add branch** button  . (Hover on the buttons in the top menu to see their names) Put it in the middle of the pane at the top, and make it the first step. Edit the properties to make it 35 by 35 and tick the **Invisible node** button.





Put two patient states below this and link them to the Branch step. Name the patient states 'tachycardia?' and 'Aged over 75 yrs?'. Now put a Sync Node below these using the **Add Sync Node** button  . Link the two patient states down to this. Edit the Sync node properties to make it 80 by 10 and turn off the Display name check box. Jin it up to a new Action box named 'Consider tachycardia in the over 75 yrs population.' It should all look like this:



Put the same GELLO in 'tachycardia?' as before. The GELLO for 'Aged over 75 yrs' is:

```
Context GLIF_VMR::GLIFDecisionNode
Let age: PQ = factory.Now() - vmr.patient.dob
Let ageInYears: PQ = age.convert('yr')

ageInYears >= factory.PQ(75, 'yr')
```

Load the test data file *GLIF_TutorialSix.xml* . Run it and see what happens. Yes the invisible node becomes invisible!

Last thing is to put some GELLO in the Sync Node.

GELLO for Sync nodes

We will use GELLO in the Sync Node to retrieve the state from the preceding patient and use some simple logic as to proceeding further.

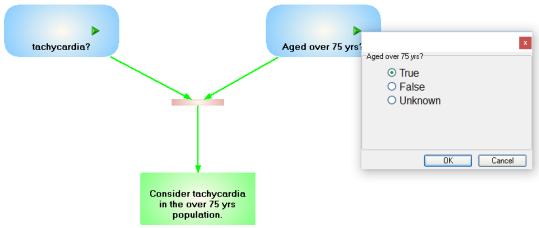
Turn off **Run**, click on on **Allow Modifications** as usual, then right click the Sync Node, **Edit properties** then **Edit GELLO**.

Use this:

```
Context GLIF_VMR::GLIFDecisionNode
Let a: Boolean = GLIFNodes.patient_state_by_name('tachycardia?')
Let b: Boolean = GLIFNodes.patient_state_by_name('Aged over 75 yrs?')

a and b
```

This will check if both the patient states are suitable as preconditions then to 'open the gate' and proceed down this path.

Here it is running with the test data. Clicking on the patient state green arrow confirms the True Boolean states.



That's the end of this tutorial.