Cooking with CQL Qs&As – Session 42

Thursday, February 27, 2020

Using Fast Healthcare Interoperability Resources (FHIR) - General

Q: Can you provide some resources for using Clinical Quality Language (CQL) in Fast Healthcare Interoperability Resources (FHIR)?

A: Here are some CQL resources:

- Getting Started with CQL (used at the January 2020 Connectathon in Baltimore MD) https://confluence.hl7.org/download/attachments/46892187/Getting%20Started%20with%20CQL.pptx?version=1&modificationDate=1555608720936&api=v2
- Community Projects – Lists all the known open source community project (discussing all the technical aspects) and implementation projects. https://github.com/cqframework/clinical_quality_language/wiki/Community-Projects
- Clinical Quality Language Project Home Page – This Health Level Seven International (HL7) Confluence page houses all of the up to date information about the project. https://confluence.hl7.org/display/CDS/Clinical+Quality+Language)
- The Clinical Decision Support (CDS) Work group will be balloting CQL Normative in the 2020 May cycle and the May Ballot plan. https://confluence.hl7.org/download/attachments/76160321/CQL%20Ballot%20Plan%20-%202020%20May.pptx?version=1&modificationDate=1582694882432&api=v2

Q: In the example on AllergyIntolerance, why are some Quality Improvement (QI)-Core Patterns, such as “allergy-active” and “allergy-confirmed,” represented in the Global Common Library?

:

```
/*
        Use of AllergyIntolerance
        */

        // connectathon/fhir4/cql/EXM105_FHIR4-8.1.000.cql
        define "Statin Allergy":
          ["AllergyIntolerance": "Statin Allergen"] StatinAllergy
            where (StatinAllergy.clinicalStatus is null or
        FHIRHelpers.ToConcept(StatinAllergy.clinicalStatus) ~ Global."allergy-active")
              and FHIRHelpers.ToConcept(StatinAllergy.verificationStatus) in {
        Global."allergy-unconfirmed", Global."allergy-confirmed" }
```

A: We define several direct reference codes related to AllergyIntolerance and other resources profiled by QI-Core in the Global Common Library so it is easy to reference. If there was a value set established, then that could be referenced directly from the Clinical Quality Language (CQL). In the interim, we reference the direct reference codes in the Global Common Library.

Q: Regarding the measure Device Indicating Frailty, when considering Fast Healthcare Interoperability Resources (FHIR)-based Clinical Quality Language (CQL) Quality Improvement (QI)-Core Patterns (http://build.fhir.org/ig/HL7/fhir-qi-core/patterns.html) within the DeviceRequest.intent comparison, is there a way to ask whether the code is "like '%order'", similar to the what is supported by some database query languages?

```
/*
        Use of DeviceRequest
        */
        // connectathon/fhir4/cql/AdvancedIllnessandFrailtyExclusion_FHIR4-4.0.000.cql
        define "Device Indicating Frailty":
          [DeviceRequest: "Frailty Device"] FrailtyDeviceOrder
            where FrailtyDeviceOrder.status in { 'active', 'on-hold', 'completed' }
              and FrailtyDeviceOrder.intent in { 'order', 'original-order', 'reflex-
        order', 'filler-order', 'instance-order' }
```

A: Yes, CQL does have that capability. There is a Matches function that supports string-based pattern matching and there are StartsWith or EndsWith functions that you can use. However, because these comparisons are terminological, the recommended approach is to define a value set that contains all and only the specific codes of interest. This example is listing the codes for illustration purposes only. Using a string-based pattern matching function for terminological comparison introduces a risk of matching on unintentional codes.