HL7® FHIR® DevDays 2019

HL7 FHIR

Using Terminology Services - Further Steps

Rob Hausam MD

Redmond, June 10 – 12 | @HL7 @FirelyTeam | #fhirdevdays | www.devdays.com/us

Who am I?

- **Name:** Rob Hausam MD
- **Company:** Hausam Consulting LLC
- **Background:**
  - Co-chair of Vocabulary and Orders and Observations WGs
  - FHIR specification and Terminology Module editor
  - Actively involved in HL7 and terminology standards/development and modeling for 17+ years

Terminology Service Operations - Overview

- ValueSet
  - $expand
  - $validate-code
- CodeSystem
  - $lookup
  - $subsumes
  - $find-matches
  - $validate-code
- ConceptMap
  - $translate
  - $closure

Renamed in R4 – previously $compose

This operation was already included in ValueSet, but is also a new addition for CodeSystem in R4

Terminology Service Operations - Overview

- ValueSet
  - $expand
  - $validate-code
- CodeSystem
  - $lookup
  - $subsumes
  - **$find-matches**
  - $validate-code
- ConceptMap
  - $translate
  - **$closure**

These operations were covered in Jim's talk yesterday

Our focus this morning
These operations so far haven't seemed to receive as much attention

Goal: Awareness, Exploration and Sharing Ideas Not Necessarily Providing Answers or Solutions

- Hope to increase awareness and stimulate interest
- Gather requirements
- Hear your thoughts and ideas

How (and Why) Would I Use $closure and $find-matches?

- First question: What are these operations?
- **$closure**
  - Provides support for ongoing maintenance of a client-side **transitive closure table** based on server-side terminological logic
    - http://build.fhir.org/conceptmap-operation-closure.html
  - See Maintaining a Closure Table
- **$find-matches**
  - Given a set of properties (and text), return one or more possible matching codes
    - http://build.fhir.org/codesystem-operation-find-matches.html

What Is Transitive Closure?

From SNOMED CT:

- The Transitive Closure is the complete set of relationships between every concept and each of its super-type concepts, in other words both its parents and ancestors

Why Would You Use Transitive Closure?

- A transitive closure table is one of the most efficient ways to test for subsumption between concepts
  - https://confluence.ihtsdotools.org/display/DOCRELFMT/4.2.5+Transitive+Closure+Files
- It can be a great way to help improve application performance when leveraging large, complex terminologies for data analysis!

Do You Need to / Want to Generate the Whole Transitive Closure Table?

- No, you don't actually have to do that
- It's often a large table and can take a long time to generate
  - Several million rows for SNOMED CT
- This is where the FHIR $closure operation can help
- Instead, you do "as needed" or "just in time" creation and maintenance of the closure table

Simple closure table

Scope	Source	Target
patient-problems	http://snomed.info/sct|22298006	http://snomed.info/sct|128599005
patient-problems	http://snomed.info/sct|24595009	http://snomed.info/sct|90560007
obs-code	http://loinc.org|14682-9	http://loinc.org|LP41281-4

Example query

```
Select * from Observations, Patients, Encounters, Conditions, Observations as Obs2 where
  Observations.patient = Patients.Key and Patients.Age > 50 and
  Observations.encounter = Encounters.Key and Encounter.clinic = [key]
    and encounter.date >= [date] and encounter.date <= [date] and
  Conditions.patient = Patients.Key and Conditions.code
    in (select Source From ClosureTable
      where Scope = "patient-problems" and Target = "http://snomed.info/sct|90560007") and
  Obs2.patient = Patients.Key and Obs2.value > 0.19 and Obs2.code
    in (select Source From ClosureTable
      where Scope = "obs-code" and Target = "http://loinc.org|LP41281-4")
```

Simple demo of the FHIR closure table

How could A FHIR closure table help with your applications?

- Your ideas here
- Questions

Moving on to $find-matches: When and how would you use it?

- This operation takes a set of properties, and examines the code system looking for codes in the code system that match a set of known properties
- 3 possible types of match
  - a complete match - a code that represents all the provided properties correctly
  - a partial match - a code that represents some of the provided properties correctly, and not others
  - a possible match - a code that may represent the provided properties closely, but may capture less or more precise information for some of the properties

$find-matches modes

- By a human
  - Looking for the best match for a set of properties
  - Server returns a list of complete, possible or partial matches (possibly with comments), so that the user can choose (or not) the most appropriate code
- By a machine
  - Server returns only a list of complete and partial matches, but no possible matches
  - The machine can choose a code from the list (or not) based on what properties are not coded
- Differentiated by the 'exact' parameter

$find-matches issues

- Very Limited server support and experience
- Unclear use cases (so far)
- Maturity level 0
- Lack of significant expressed interest to move the maturity forward

Next Steps

- Add these operations to upcoming FHIR Connectathon Terminology Service tracks
- Gain experience and feedback
- Let us know your thoughts and needs
- For $find-matches, if we can't move beyond draft for R5 – consider removing from specification?