2020-02-26 - SLPG Meeting

Date & Time
20:00 UTC Wednesday 26th February 2020

Goals
- To identify steps needed to finalize URI updates
- To finalize requirements for term searching in ECL
- To further discuss query syntax for accessing reference sets

Location
Zoom meeting: https://snomed.zoom.us/j/471420169

Attendees
- Chair: Linda Bird
- Project Group: Daniel Karlsson, Ed Cheetham, Michael Lawley, Peter Jordan, Rob Hausam, Anne Randorff Højen

Apologies

Agenda and Meeting Notes

<table>
<thead>
<tr>
<th>Description</th>
<th>Owner</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome and agenda</td>
<td>Linda Bird</td>
<td>Please note that the SLPG will be meeting in London on Sunday 5th April (9am to 12:30pm) - see schedule</td>
</tr>
<tr>
<td>Concrete values</td>
<td>Linda Bird</td>
<td>SCG, ECL, STS, ETL - Ready for publication - on hold until after MAG meeting in April confirming requirement for Boolean datatype.</td>
</tr>
</tbody>
</table>
| URIs                 | Peter G. Williams & Linda Bird | Please review updates to the URI specification, and raise any comments in the next 2 weeks. Draft URI standard for review - URI Standard

- 2.6 URIs for Language Syntaxes
- 2.7 URIs for Language Instances
- 2.8 URIs for Modelling Resources
- 3.1 Resolving SNOMED CT URIs

Expression Constraint Language

<table>
<thead>
<tr>
<th>Description</th>
<th>Owner</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>NEXT STEP FOR ECL:</td>
<td>Linda Bird</td>
<td>Agreement in Malaysia - ECL will add the following term searching syntax (no regex - just wild card and word prefix any order):</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{term} &= [ \text{termSearchType} : \text{String}, \text{language} : \text{<langCode>} ] \\
\text{Example} &= \{ \{ \text{term} = \text{"heart att"}, \text{language} = \text{en} \} \}
\end{align*}
\]

- Question - Do we want to reconsider including optional parameters for ‘type’, ‘dialect’ and ‘acceptability’
  - typeId = 900000000000013009 ; type = <synonym | fsn>
  - dialectId = 900000000000508004 ; dialect = <en-GB | en-AU | en-Patient | de-CardioSpecialist>
  - dialectId = 900000000000508004 + 900000000000509007 ; dialect = en-GB + en-US
  - acceptabilityId = 900000000000549004 ; acceptability = <acceptable | preferred>

Term Search Type

a. Wild Card Match (collation) - e.g.
   - \{ \{ \text{term} = \text{wild} : "heart*" \} \}
   - \{ \{ \text{term} = \text{wild (sv)}:="hjärta*" \} \}
a. Word Prefix Any Order - e.g.
   - \{ \{ \text{term} = \text{match} : "hear att" \} \}
a. Default (word prefix any order) - e.g.
   - \{ \{ \text{term} = \text{"hear att"} \} \}
   - \{ \{ \text{term} = \text{"heart"} \} \}

Potential Examples
Use Cases

- Intentionally define a reference set for chronic disease. Starting point was ECL with modelling; This misses concepts modelled using the pattern you would expect. So important in building out that reference set.
- Authors quality assuring names of concepts
- Checking translations, retranslating. Queries for a concept that has one word in Swedish, another word in English
- AU use case would have at most 3 or 4 words in match
- Consistency of implementation in different terminology services
- Authoring use cases currently supported by description templates
- A set of the “ectomy’s” and “itis’s”

Questions

- Do we include ‘typeId’ - e.g. << 64572001 |Disease| {{ D.term = “heart”; typeId = 900000000000013009 |Synonym| }}
  - NO
- Do we include ‘type’ - e.g. << 64572001 |Disease| {{ D.term = “heart”; D.type = synonym }}
  - NO
- Do we include ‘languageCode’ - e.g. << 64572001 |Disease| {{ D.term = “heart”; D.type = synonym, D.languageCode = “en” }}
  - YES
- Do we include ‘caseSignificanceCancelId’ - e.g. << 64572001 |Disease| {{ D.term = “Heart”; D.caseSignificanceCancelId = 900000000000017005 |case sensitive| }}
  - NO
- Do we include ‘caseSignificance’ - e.g. << 64572001 |Disease| {{ D.term = “Heart”; D.caseSignificance = sensitive }}
  - NO
- Do we include ‘language’ and ‘version’ - e.g. << 64572001 |Disease| {{ term = “heart”; version = http://… |LANGUAGE = (999001881000000108|Gastro LRS|, |GB English|) }}
  - NO
- Do we include syntactic sugar - e.g.
  - << 64572001 |Disease| {{ preferredTerm = “heart”; languageRefSet = en-gb}}
  - << 64572001 |Disease| {{ fullySpecifiedTerm = “heart”; languageRefSet=en-gb}}
  - << 64572001 |Disease| {{ acceptableTerm = “heart”; languageRefSet=en-gb}}
  - << 64572001 |Disease| {{ preferredTerm = “heart” FROM version = X, language = Y }}
  - NO
- Do we use/require the “D” at the start of “term”?  
  - NO
- Packaging - How do we package this extension to ECL
  - A new version of ECL - version 1.5

Querying Refset Attributes

<table>
<thead>
<tr>
<th>Linda Bird</th>
<th>Proposed syntax to support querying and return of alternative refset attributes (To be included in the SNOMED Query Language)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Example use cases</strong></td>
</tr>
<tr>
<td></td>
<td>- Execution of maps from international substance concepts</td>
</tr>
<tr>
<td></td>
<td>- Find the anatomical parts of a given anatomy structure concept (in Anatomy structure and part association reference set)</td>
</tr>
<tr>
<td></td>
<td>- Find potential replacement concepts for an inactive concept in record</td>
</tr>
<tr>
<td></td>
<td>- Find the order of a given concept in an Ordered component reference set</td>
</tr>
<tr>
<td></td>
<td>- Find a concept with a given order in an Ordered component reference set</td>
</tr>
</tbody>
</table>
Potential syntax to consider (brainstorming ideas)

- **SELECT ??**
  - SELECT 123 |referenced component|, 456 |target component|
  - FROM 799 |Anatomy structure and part association refset|
  - WHERE 123 |referenced component| = (< 888 |Upper abdomen structure| {{ term = "heart" }} )
- SELECT id, moduleId
  - FROM concept
  - WHERE id IN (< |Clinical finding|)
  - AND definitionStatus = |primitive|
- SELECT ??? |id|, ??? |moduleId|
  - FROM concept, ECL("< |Clinical finding") CF
  - WHERE concept.id = CF.sctid
  - AND definitionStatus = |primitive|
- **SELECT ?? |id|, ?? |moduleId|
  - FROM concept ( < |Clinical finding| {{ term = "heart" }} {{ definitionStatus = |primitive| }} )

**Question** - Can we assume some table joins - e.g. Concept.id = Description.conceptId etc ??

**Examples**
- Try to recast relationships table as a Refset table + graph-based extension
- Find primitive concepts in a hierarchy

**ROW ... ?**
- ROWOF (|Anatomy structure and part association refset|) ? (|target ComponentId|
  - same as: ^ |Anatomy structure and part association refset|
  - ROWOF (|Anatomy structure and part association refset|) . |target ComponentId|
- ROWOF (< 900000000000496009|Simple map type reference set| {{ term = "my hospital"}) {{ 449608002|Referenced component| = 80581009 |Upper abdomen structure|})
  - 900000000000505001 |Map target|
  - (ROW (< 900000000000496009|Simple map type reference set| {{ term = "my hospital"}) : 449608002|Referenced component| = 80581009 |Upper abdomen structure|).
  - 900000000000505001 |Map target|

**# ... ?**
- # |Anatomy structure and part association refset| ? |referenced component|
- # |My ordered component refset| |Referenced component| = |Upper abdomen structure| ? |priority order|
- ? |My ordered component refset| {{ |Referenced component| = |Upper abdomen structure| }} . |priority order|
- ? |My ordered component refset| . |Referenced component|
- equivalent to ^ |My ordered component refset|
- ? (<|My ordered component refset|) {{ |Referenced component| = |Upper abdomen structure| }} . |priority order|
- ? (<|My ordered component refset|) {{ term = "map"}) ) |( |Referenced component| = |Upper abdomen structure| ) . |priority order|
- REFSETROWS (<|My ordered component refset|) {{ term = "map"}) ) |( |Referenced component| = |Upper abdomen structure| ) SELECT |priority order|

**Specify value to be returned**
- 7 449608002 |Referenced component|?
  - 734139008 |Anatomy structure and part association refset|)
  - ^ 734139008 |Anatomy structure and part association refset| (Same as previous)
  - 7 900000000000533001 |Association target component|?
  - 7 734139008 |Anatomy structure and part association refset|)
  - 9 900000000000533001 |Association target component|?
  - 7 734139008 |Anatomy structure and part association refset|)
  - 9 900000000000533001 |Association target component|?
  - 7 734139008 |Anatomy structure and part association refset|)
  - 9 900000000000533001 |Association target component|?
  - 7 734139008 |Anatomy structure and part association refset|)
  - 9 900000000000533001 |Association target component|?
  - 7 734139008 |Anatomy structure and part association refset|)
  - 9 900000000000533001 |Association target component|?
  - 7 449608002 |Referenced component| = (< |Upper abdomen structure| )
  - 7 900000000000533001 |Association target component|?
  - 7 734139008 |Anatomy structure and part association refset|)
  - 9 449608002 |Referenced component| = (< |Upper abdomen structure|) : |Finding site| = *)


<table>
<thead>
<tr>
<th>Returning Attributes</th>
<th>Michael Lawley</th>
<th>Proposal (by Michael) for discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• Currently ECL expressions can match (return) concepts that are either the source or the target of a relationship triple (target is accessed via the ‘reverse’ notation or ‘dot notation’, but not the relationship type (ie attribute name) itself. For example, I can write:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;= 404684003[Clinical finding] . 363698007[Finding site]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>But I can’t get all the attribute names that are used by &lt;= 404684003[Clinical finding]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Perhaps something like:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ? R.type ? (&lt;= 404684003 [Clinical finding])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• This could be extended to, for example, return different values - e.g.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ?</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reverse Member Of</th>
<th>Michael Lawley</th>
<th>Proposal for discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>What refsets is a given concept (e.g. 421235005 [Structure of femur]) a member of?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Possible new notation for this:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ^ . 421235005 [Structure of femur]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• ? X ? 421235005 [Structure of femur] = ^ X</td>
</tr>
</tbody>
</table>
ON HOLD WAITING FROM IMPLEMENTATION FEEDBACK FROM INTERNAL TECH TEAM

WIP version - https://confluence.ihtsdotools.org/display/WIPSTS/Template+Syntax+Specification
- Added a 'default' constraint to each replacement slot - e.g. default (72673000 |Bone structure (body structure))
- Enabling 'slot references' to be used within the value constraint of a replacement slot - e.g. [[ +id (<< 123037004 |Body structure (body structure)|) MINUS $findingSite2] @findingSite1]
- Allowing repeating role groups to be referenced using an array - e.g. $rolegroup[1] or $rolegroup[[SELF]]
- Allowing reference to 'SELF' in role group arrays
- Adding 'sameValue' and 'allOrNone' constraints to information slots - e.g. sameValue ($site), allOrNone ($occurrence)

See changes in red here: 5.1. Normative Specification

Examples:

[[+id]]: [[1..*] @my_group sameValue(morphology)] ( [Finding site] = [[ +id (<<123037004 |Body structure (body structure)|) MINUS $site[! SELF ] ] @site ] , [Associated morphology] = [[ +id @my_morphology ]] )

- Implementation feedback on draft updates to Expression Template Language syntax
- Use cases from the Quality Improvement Project:
  - Multiple instances of the same role group, with some attributes the same and others different. Eg same morphology, potentially different finding sites.

Note that QI Project is coming from a radically different use case. Instead of filling template slots, we're looking at existing content and asking "exactly how does this concept fail to comply to this template?"

For discussion:

[[0..1] ( [[0..1]] 246075003 |Causative agent| = [[+id (< 410607006 |Organism| ) @Organism]]) ]

Is it correct to say either one of the cardinality blocks is redundant? What are the implications of 1..1 on either side? This is less obvious for the self grouped case.

Road Forward for SI

1. Generate the parser from the ABNF and implement in the Template Service
2. User Interface to a) allow users to specify template at runtime b) tabular (auto-completion) lookup STL
3. Template Service to allow multiple templates to be specified for alignment check (aligns to none-off)
4. Output must clearly indicate exactly what feature of concept caused misalignment, and what condition was not met.

Additional note: QI project is no longer working in subhierarchies. Every 'set' of concepts is selected via ECL. In fact most reports should now move to this way of working since a subhierarchy is the trivial case. For a given template, we additionally specify the "domain" to which it should be applied via ECL. This is much more specific than using the focus concept which is usually the PPP eg Disease.

FYI Michael Chu
### Query Language - Summary from previous meetings

**Query Language - Summary from previous meetings**

**Query Language - Summary from previous meetings**

<table>
<thead>
<tr>
<th>Query Language - Summary from previous meetings</th>
<th>Linda Bird</th>
<th>FUTURE WORK</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Examples: version and dialect</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• &lt;&lt; 64572001</td>
<td>Disease</td>
<td>([ term = &quot;&quot;heart&quot;&quot; ])) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></td>
</tr>
<tr>
<td>• &lt;&lt; 64572001</td>
<td>Disease</td>
<td>([ synonym = &quot;&quot;heart&quot;&quot; ])) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></td>
</tr>
<tr>
<td>• &lt;&lt; 64572001</td>
<td>Disease</td>
<td>([ FSN = &quot;&quot;heart&quot;&quot; ])) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, DIALECT W</td>
</tr>
<tr>
<td>• &lt;&lt; 64572001</td>
<td>Disease</td>
<td>([ preferredTerm = &quot;&quot;heart&quot;&quot; ])) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, DIALECT Y</td>
</tr>
<tr>
<td>• (* ([ term = &quot;&quot;heart&quot;&quot; ])) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, DIALECT Z</td>
<td>MINUS (* ([ term = &quot;&quot;heart&quot;&quot; ])) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20170731">http://snomed.info/sct/900000000000207008/version/20170731</a>, DIALECT W</td>
<td></td>
</tr>
<tr>
<td>X MINUS Y WHERE X = &quot;&quot;, Y = ( * ([ term = &quot;&quot;heart&quot;&quot; ])) VERSION <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, DIALECT W</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Notes</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Allow nested where, version, language</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Scope of variables is inner query</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| Examples: where                                 |            |             |
| • X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547) |            |             |
| • X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547) VERSION http://snomed.info/sct/900000000000207008/version/20180131 |
| • X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547), Y = (<< 1456) VERSION http://snomed.info/sct/900000000000207008/version/20180131 |
| • X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547) VERSION http://snomed.info/sct/900000000000207008/version/20180131, LANGUAGE 900000000000508004 [GB English] |
| • X MINUS >! X WHERE X = (<< 1234 : 5678 = << 6547) VERSION http://snomed.info/sct/900000000000207008/version/20180131, LANGUAGE 999001881000000108[GB clinical extension LRS], 900000000000508004 [GB English] |
| • X minus >! X WHERE M = (<< 1234)) VERSION http://snomed.info/sct/900000000000207008/version/20180131, DIALECT 999001881000000108[GB clinical extension LRS], 900000000000508004 [GB English] |
| **Notes**                                       |            |             |
| • Allow nested variable definitions, but recommend that people don’t due to readability |
| • Scope of variables is the inner query        |            |             |
| • No recursion e.g X WHERE X = 1234 MINUS X    |            |             |
| • ie can’t use a variable in its own definition |
| • ie X is only known on the left of the correspondig WHERE, and not on the right of the WHERE |
Keywords for Term-based searching:

- **D.term**
  - D.term = "*heart"
  - D.term = wild: "*heart"
  - D.term = regex: "*heart"
  - D.term = match: "hear att"
  - D.term = (sv) wild: "*heart"

- **D.languageCode**
  - D.languageCode = "en"
  - D.languageCode = "es"

- **D.caseSignificance**
  - D.caseSignificance = "entire term case insensitive"
  - D.caseSignificance = "entire term case sensitive"
  - D.caseSignificance = "only initial character case insensitive"

- **D.type**
  - D.type = "FSN"
  - D.type = fullySpecifiedName
  - D.type = "synonym"
  - D.type = definition

- **D.acceptability**
  - D.acceptability = "acceptable"
  - D.acceptability = "preferred"

Additional Syntactic Sugar

- **FSN**
  - FSN = "*heart"
  - D.type = "FSN"
  - D.type = fullySpecifiedName

- **synonym**
  - synonym = "*heart"
  - D.type = synonym

- **syntaxOrFSN**
  - synonymOrFSN = "*heart"
  - D.type = synonym OR D.type = fullySpecifiedName

- **textDefinition**
  - textDefinition = "*heart"

Unacceptable Terms

- (D.term = "*heart") MINUS (D.term = "*heart", D.acceptability = "en")
Language preferences using multiple language reference sets

- LRSs that use the same Language tend to use 'Addition' - i.e. child LRS only includes additional acceptable terms, but can override the preferred term
  - E.g. Regional LRS that adds local dialect to a National LRS
  - E.g. Specialty-specific LRS
  - E.g. Irish LRS that adds local preferences to the en-GB LRS
    - 99999900 {Irish language reference set} PLUS [GB English reference set]
- LRSs that define a translation to a different language tend to use 'Replacement' - i.e. child LRS replaces set of acceptable and preferred terms for any associated concept
  - E.g. Danish LRS that does a partial translation of the International Release
    - 999999 {Danish language reference set} ELSE [GB English reference set]

Next meeting is scheduled for Wednesday 11th March 2020 at 20:00 UTC.

No files shared here yet.