### 2018-06-06 - SLPG Meeting

**Date & Time**
20:00 UTC Wednesday 6th June 2018

**Teleconference Details**
To join the meeting please go to https://snomed.zoom.us/j/471420169.
Further information can be found at SLPG meeting information

**Goals**
- Progress SNOMED Query language
- Discuss use of multiple language reference sets
- URI standard
  - URI for canonical normal form
  - URI for necessary (long/short) normal form
- Future goals:
  - Transitive relationships in ECL
  - Ability to execute maps from within ECL

**Attendees**
- Chair: Linda Bird
- Project Group: Michael Lawley, Ed Cheetham, Rob Hausam

**Apologies**
Anne Randorff Højen

## Agenda and Meeting Notes

<table>
<thead>
<tr>
<th>Description</th>
<th>Owner</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome and apologies</td>
<td>Linda Bird</td>
<td></td>
</tr>
<tr>
<td>Query Language - Recap from previous meetings</td>
<td>Linda Bird</td>
<td></td>
</tr>
</tbody>
</table>

### Examples: version and language
- ```<< 64572001 [Disease] {{ term = "*heart*" }} VERSION http://snomed.info/sct/900000000000207008/version/20180131
  LANGUAGE X MINUS Y WHERE X = *, Y = (* {{ term = "*heart*" }}) VERSION http://snomed.info/sct/900000000000207008/version/20170731, LANGUAGE W

#### Notes
- Allow nested where, version, language
- Scope of variables is inner query```
### 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 >! Y 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 X = ( < M WHERE M = (<< 1234)) VERSION http://snomed.info/sct/900000000000207008/version/20180131, LANGUAGE 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 corresponding WHERE, and not on the right of the WHERE`
Query Language - Recap from previous meetings

<table>
<thead>
<tr>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td>What filter keywords will we introduce for Term-based searching, and what are their exact meanings?</td>
</tr>
</tbody>
</table>

- **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.caseSignificanceId**
  - D.caseSignificanceId = 900000000000448009 |entire term case insensitive|
  - D.caseSignificanceId = 900000000000017005 |entire term case sensitive|
  - D.caseSignificanceId = 900000000000020002 |only initial character case insensitive|

- **D.caseSignificance**
  - D.caseSignificance = "insensitive"
  - D.caseSignificance = "sensitive"
  - D.caseSignificance = "initialCharInsensitive"

- **D.typeId**
  - D.typeId = 900000000000003001 |fully specified name|
  - D.typeId = 900000000000013009 |synonym|
  - D.typeId = 900000000000550004 |definition|

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

- **D.acceptabilityId**
  - D.acceptabilityId = 900000000000549004 |acceptable|
  - D.acceptabilityId = 900000000000548007 |preferred|

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

### Additional Syntactic Sugar

- **FSN**
  - FSN = "*heart"
    - D.term = "*heart", D.type = "FSN"
    - FSN = "*heart" LANGUAGE X

- **synonym**
  - synonym = "*heart"
    - D.term = "*heart", D.type = "synonym"
    - synonym = "*heart" LANGUAGE X

- **synonymOrFSN**
  - synonymOrFSN = "*heart"
    - synonym = "*heart" OR FSN = "*heart"
      - D.term = "*heart", OR D.type = "fullySpecifiedName"

- **textDefinition**
  - textDefinition = "*heart"
    - textDefinition = "*heart" LANGUAGE X

### Unacceptable Terms

- (D.term = "*heart") MINUS (D.term = "*heart", D.acceptability = "acceptable" OR D.acceptability = "preferred") LANGUAGE X
<table>
<thead>
<tr>
<th>Query Language - Combining language reference sets</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td>How do we support language preferences, which are defined over multiple language reference sets? For example:</td>
<td></td>
</tr>
<tr>
<td>• 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</td>
<td></td>
</tr>
<tr>
<td>• E.g. Regional LRS that adds local dialect to a National LRS</td>
<td></td>
</tr>
<tr>
<td>• E.g. Specialty-specific LRS</td>
<td></td>
</tr>
<tr>
<td>• E.g. Irish LRS that adds local preferences to the en-GB LRS</td>
<td></td>
</tr>
<tr>
<td>• 99999900 [Irish language reference set] PLUS [GB English reference set]</td>
<td></td>
</tr>
<tr>
<td>• 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</td>
<td></td>
</tr>
<tr>
<td>• E.g. Danish LRS that does a partial translation of the International Release</td>
<td></td>
</tr>
<tr>
<td>• 999999 [Danish language reference set] ELSE [GB English reference set]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>URI Specification</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Status update</td>
<td></td>
</tr>
<tr>
<td>• URLs for canonical necessary normal form and necessary (long/short) normal form</td>
<td></td>
</tr>
<tr>
<td>• Next Meeting:</td>
<td></td>
</tr>
<tr>
<td>• Recap on purpose of SNOMED CT computable language URIs?</td>
<td></td>
</tr>
<tr>
<td>• Recap on language instance URIs (e.g. URIs for expressions and expression constraints)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Other topics</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td>Other topics for discussion. For example:</td>
<td></td>
</tr>
<tr>
<td>• Query language - Can we de-scope relationship filters?</td>
<td></td>
</tr>
<tr>
<td>• ECL suggestions - Ability to execute maps in ECL</td>
<td></td>
</tr>
<tr>
<td>• The specific use-case here comes initially from Jeremy and relates to being able to work with inactive concepts via the historical association maps. For example, given an ECL expression, e, that identifies a set of concepts to be used for retrieving patient records, you probably also want to retrieve records for sameAs(e) and replacedWith(e)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Confirm next meeting date /time</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td>The next SLPG meeting will be held in 2 weeks at 20:00 UTC on <strong>Wednesday 4th July</strong>.</td>
<td></td>
</tr>
</tbody>
</table>