2018-03-28 - SLPG Meeting

Date & Time
20:00 UTC Wednesday 28th March 2018

Goals
• Clarify execution semantics of reverse cardinality in ECL
• Progress SNOMED Query language

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

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

Apologies

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>ECL reverse cardinality</td>
<td>Linda Bird</td>
<td>Question</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------</td>
<td>----------</td>
</tr>
<tr>
<td>I have a doubt in expression constraints with reverse cardinalities: only non-redundant attributes have to be included in the cardinality count?</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

For example, the following EC:

```
```

returns this concept (among others): 72551001 |Metal fumes (substance)|. This substance is the causative agent of three clinical findings: 74800004 |Brass-founders' fever|, 72163003 |Metal fever| and 308905009 |Welders ague|, and 74800004 |Brass-founders' fever| and 308905009 |Welders ague| are a type of 72163003 |Metal fever|. So the question is: does the EC above has to return 72551001 |Metal fumes (substance)|? Or this substance should be matched with [2..x] cardinality? (Please note that x corresponds to 2, 3...*).

**Proposed Answer** (for discussion)

At the moment, existing implementations (that I know of) seem to return 72551001 |Metal fumes (substance)|; and the SNOMED International Browser shows the 3 concepts that you mention in the list of causative agent "References" for the concept 72163003 |Metal fever|.

However, whether or not this is the correct semantics for ECL comes down to how we interpret "redundant relationships". I think this may require a further discussion with the implementation community (which I can start through the SNOMED Languages Project Group). However, these are my current thoughts on how redundancy should be treated.... Given concepts X and Y, and attribute R:

a. IF relationship (X R Y) (i.e. "All X have an R relationship with some Y") THEN relationship (X R supertype(Y)) (i.e. "All X have an R relationship with some supertype of Y")

b. IF relationship (X R Y) (i.e. "All X have an R relationship with some Y") THEN relationship (subtype(X) R Y) (i.e. "All subtypes of X have an R relationships with some Y")

An example of the first inference is:

- IF relationship ([Copper fever],[Causative agent], [Metal fumes]) THEN relationship ([Copper fever], [C]ausative agent, [Metal fumes])

Therefore if both these relationships are true, then relationship ([Copper fever], [Causative agent], [Metal fumes]) is redundant

An example of the second inference is:

- IF relationship ([Metal fever],[Causative agent], [Metal fumes]) THEN relationship ([Brass-founders' fever], [Causative agent], [Metal fumes])

Therefore if both these relationships are true, then relationship ([Brass-founders' fever], [Causative agent], [Metal fumes]) is redundant

If this logic above is correct, then the only non-redundant relationship involving a [Causative agent] of [Metal fumes] is the relationship ([Metal fever], [Causative agent], [Metal fumes]) ... in which case the concept [Metal fumes] should only be returned when the cardinality include [1]. For example:

```
<< 10559001 [substance]: [1..1] R 246075003 [causative agent] = << 404684003 [clinical finding]
```

However, I would like to discuss this further with the implementation community, given this is not the way current implementation are working.
<table>
<thead>
<tr>
<th>Query Language - Recap from last week</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Examples: version and language</strong></td>
<td></td>
</tr>
<tr>
<td>- `&lt;&lt; 64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>- `&lt;&lt; 64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>- `&lt;&lt; 64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>- `&lt;&lt; 64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>- `&lt;&lt; 64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>- `&lt;&lt; 64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>- `&lt;&lt; 64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>- `&lt;&lt; 64572001</td>
<td>Disease</td>
</tr>
<tr>
<td>- `X MINUS Y WHERE X = &quot;&quot;, Y = &quot;{&quot;</td>
<td>VERSION <a href="http://snomed.info/sct/900000000000207008/20180131">http://snomed.info/sct/900000000000207008/20180131</a>, LANGUAGE W`</td>
</tr>
</tbody>
</table>

**Notes**

- Allow nested where, version, language
- Scope of variables is inner query

<table>
<thead>
<tr>
<th>Examples: where</th>
</tr>
</thead>
<tbody>
<tr>
<td>- <code>X MINUS : X WHERE X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547)</code></td>
</tr>
<tr>
<td>- <code>X MINUS : X WHERE X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547) VERSION http://snomed.info/sct/900000000000207008/20180131</code></td>
</tr>
<tr>
<td>- <code>X MINUS : Y WHERE X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547), Y = (&lt;&lt; 1456) VERSION http://snomed.info/sct/900000000000207008/20180131</code></td>
</tr>
<tr>
<td>- <code>X MINUS : X WHERE X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547) VERSION http://snomed.info/sct/900000000000207008/20180131, LANGUAGE 9000000000000508004 [GB English]</code></td>
</tr>
<tr>
<td>- <code>X MINUS : X WHERE X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547) VERSION http://snomed.info/sct/900000000000207008/20180131, LANGUAGE 999001881000000108 [GB clinical extension LRS], 9000000000000508004 [GB English]</code></td>
</tr>
<tr>
<td>- <code>X minus : X WHERE X = ( &lt; M WHERE M = (&lt;&lt; 1234)) VERSION http://snomed.info/sct/900000000000207008/20180131, LANGUAGE 999001881000000108 [GB clinical extension LRS], 9000000000000508004 [GB English]</code></td>
</tr>
</tbody>
</table>

**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`
How do we support language preferences, which are defined over multiple language reference sets?

For example:

- **Assume:** No concept has descriptions in 2 listed language refsets ...
- **But if they do:** do they override or are they additive?
- **Option 1:** PTs override (so order is important), and acceptable terms are additive. If there are 2 PTs, then the second PT is acceptable.
  - See, for example 999001891000000105 [Paediatric neurodisability outpatient diagnosis language reference set]
- **Option 2:** Use '+' to indicate additive, and ',' to indicate priority order (with override). For example:
  - LANGUAGE 999001891000000105 [Paediatric neurodisability outpatient diagnosis language reference set], 900000000000508004 [GB English]
    - Priority order: This means that if a concept has descriptions in the Paediatric LRS, then this LRS is used. But if a concept has no descriptions in the Paediatric LRS, then GB English is used
  - LANGUAGE 999001881000000108[GB clinical extension LRS] + 900000000000508004 [GB English]
    - Additive: This means that if a concept has a PT in both LRSs, then the PTs in the Paediatric LRS take priority, and the PT in the GB English LRS becomes acceptable. Other terms are acceptable if they are acceptable in either LRS.
  - LANGUAGE 999001891000000105 [Paediatric neurodisability outpatient diagnosis language reference set], 999001881000000108[GB clinical extension LRS] + 900000000000508004 [GB English]
    - Priority order and Additive: This means that if a concept has a PT in both LRSs, then the PTs in the Paediatric LRS take priority, and the PT in the GB English LRS becomes acceptable. Other terms are acceptable if they are acceptable in either LRS.
### Filters for Lexical Searching

<table>
<thead>
<tr>
<th>Linda Bird</th>
</tr>
</thead>
</table>

**What filter keywords will we introduce for Term-based searching, and what are their exact meanings?**

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

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

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

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

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

### Confirm next meeting date / time

Linda Bird

Due to the April SNOMED business meeting in London, the next SLPG meeting will be held in 4 weeks at 20:00 UTC on **Wednesday 25th April 2018**.