Date & Time
20:00 UTC Wednesday 27th March 2019

Goals
- Review actions from last meeting
- Proposed enhancements to template language
- Proposed new language features for mapping

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, Anne Randorff Højen, Yongsheng Gao

Apologies
- Daniel Karlsson

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>Actions from last week</td>
<td>Linda Bird</td>
<td>- Actions from last week:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Consider new syntax to support proposed expression template use case</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Add examples with 2-levels of nesting</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Revise notation based on last meeting's discussion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Consider new syntax to support proposed map use case</td>
</tr>
<tr>
<td>Template Syntax</td>
<td>Linda Bird</td>
<td>Use cases: New concept development, querying based on template matching, and template-based modeling transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>New requirements</td>
</tr>
</tbody>
</table>
1. Constrain values across 2 or more replacement slots
   - 2 replacement slots must have the same value, different values, subsumed values, or not subsumed values.
   - **Example A** - A clinical finding, with 2 role groups with the same morphology, and finding sites that do not subsume each other
     - **Template**
       - `[[ +id ]]:
         { 116676008 |Associated morphology| = [[ +id @morphology ]],
           363698007 |finding site| = [[ +id (<< Body structure] MINUS << $findingSite2 ) @findingSite1 ]],
           116676008 |Associated morphology| = [[ +id @morphology ]],
           363698007 |finding site| = [[ +id (* MINUS << $findingSite1) @findingSite2 ]] )
     - **Valid Expression** (Definition of 1602739100119109 [Bone cyst of bilateral tibias (disorder)])
       - 64572001 |Disease|:
         { 116676008 |Associated morphology| = 66954000 |Bone cyst|,
           363698007 |Finding site| = 719491009 |Bone structure of right tibia|,
           116676008 |Associated morphology| = 66954000 |Bone cyst|,
           363698007 |Finding site| = 719492002 |Bone structure of left tibia|)
   - **Example B** - A clinical finding, with one or more role groups in which the morphology is always the same, and no 2 finding sites subsume each other.
     - **Template** - 3 role groups with 3 sites: site[1], site[2], site [3] /// site [1,2]
     - `[[[+id]]]:
       [[[.+!*] @group1] |Finding site| = [[ +id (* MINUS << @site )] ,
       |Associated morphology| = [[ +id ($morphology ! SELF )] @morphology ] ,
       Valid Expression (Definition of 31580001000004106 [Bilateral sacral insufficiency fracture (disorder)])
     - 702561005 |Insufficiency fracture (disorder)|:
       { 363698007 |Finding site (attribute)| = 736830008 |Bone structure of left half of sacrum (body structure)|,
       116676008 |Associated morphology (attribute)| = 22640007 |Pathologic fracture (morphologic abnormality)| )
       { 363698007 |Finding site (attribute)| = 736831007 |Bone structure of right half of sacrum (body structure)|,
       116676008 |Associated morphology (attribute)| = 22640007 |Pathologic fracture (morphologic abnormality)| )

2. Default value for replacement slot
   - Default value for authoring and template-driven modelling transformations
   - **Example A** - Default finding site of 72673000 |Bone structure|
     - **Template**
       - `[[ + id ]]:
         { |Finding site| = [[ +id (<< 72673000|Bone structure|) @site default (72673000 |Bone structure (body structure)|) ]]

3. Definition status of a replacement slot
   - Specifying whether the value used in a replacement slot must be primitive or defined
   - **Example A** - When proximal primitive modelling, the focus concept must be a primitive concept
     - **Template** - Requires use of more expressive query language with filters
       - `[[ + id (<< 64572001 |Disease| [[ c.definitionStatus = primitive ]]) ]]
     - **Valid Expression**
       - 195967001 |Asthma (disorder)|

4. Definition status of a templated expression
   - Specifying the definition status of a templated expression
     - **Template**
       - `[[ +tok ("===" "<<<") ][+id ]]: { |Finding site| = [[+id]] }
     - **Valid Expression**
       - `=== 128272009 [Disorder of lower respiratory system]|: 363698007 |Finding site| = 39607008 |Lung structure|
       - <<< 128272009 [Disorder of lower respiratory system]|: 363698007 |Finding site| = 39607008 |Lung structure|

5. Attributes used in repeating role groups
   - Constraining the set of attributes that appear in a repeating role group
**Example 1** - In a given matching expression, either all the role groups include the attribute |site|, or none of the role groups include the attribute |site|. Similarly, either all role groups include |Occurrence|, or none of the role groups do.

**Template - using allOrNone**

- 

- Valid Expression - Injury of head, neck and chest

  - Valid Expression - Congenital malformation of head and neck
Example 2 - Some of the optional attributes must either always or never appear in each instance of a repeating role group

- Valid Expression - Closure of skin by suture
  - Procedure:
    - [Method] = [Closing - action], [Procedure site - Direct] = [Skin structure], [Using device] = [Surgical suture, device]

- Valid Expression - Core needle biopsy of skin using ultrasound guidance
  - Procedure:
    - [Method] = [Ultrasound imaging - action], [Procedure site - Direct] = [Skin structure], [Using device] = [Core biopsy needle, device]

Example 3 - Some of the optional attributes must either always or never appear in each instance of an inner-nested, repeating role group

- Valid Expression - History of injury of head, neck and chest, and of congenital malformation of head and neck
  - Finding with explicit context:
    - [Finding context] = [Done], [Temporal context] = [In the past], [Associated finding] = [Disease]:
      - [Associated morphology] = [Injury], [Finding site] = [Head structure]
      - [Associated morphology] = [Injury], [Finding site] = [Neck structure]
      - [Associated morphology] = [Injury], [Finding site] = [Chest structure]

Example 4 - Some of the optional attributes must either always or never appear in each instance of an outer-nested, repeating role group

- Valid Expression - History of injury of head, neck and chest, and of malformation of head and neck
  - Finding with explicit context:
    - [Finding context] = [Done], [Temporal context] = [In the past], [Associated finding] = [Disease]:
      - [Associated morphology] = [Injury], [Finding site] = [Head structure]
      - [Associated morphology] = [Injury], [Finding site] = [Neck structure]
      - [Associated morphology] = [Injury], [Finding site] = [Chest structure]
### Executing maps

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

Linda Bird has proposed an extension to ECL to support the execution of maps.

- **Example use cases**
  - Mapping from international substance concepts to AMT substance concepts
  - Anatomy structure and part association reference set - e.g. find the anatomical parts of a given structure

- **Potential syntax to consider**
  - **Functional**
    - `mapTarget (\{Anatomy structure and part association refset\}, << |Upper abdomen structure|)`
    - `mapSource (\{Anatomy structure and part association refset\}, << |Liver part|)`
  - **Dot notation**
    - `|Anatomy structure and part association refset| . |mapTarget|`
  - **Filters**
    - `\{ |Anatomy structure and part association refset| \} . |mapTarget|`

### Returning attributes

<table>
<thead>
<tr>
<th>Michael Lawley</th>
</tr>
</thead>
</table>

Michael Lawley has proposed a feature for retrieving attribute names.

- **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:

  ```
  << 404684003|Clinical finding| : 363698007|Finding site| = <<66019005|Limb structure|
  << 404684003|Clinical finding| . 363698007|Finding site|
  ```

  But I can't get all the attribute names that are used by `<< 404684003|Clinical finding|`

  - **Perhaps something like:**
    - `? R.type ? (<< 404684003 |Clinical finding|)`
    - **This could be extended to, for example, return different values - e.g.**
      - `?? |Simple map refset|.|maptarget| ?? (^|Simple map refset| AND < |Fracture|)`

### URI Standard

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

- **Finalize and publish language and language instance URIs**
**Query Language - Summary from previous meetings**

**Examples: version and language**

- `<< 64572001 |Disease| [{term = "*heart*"}] VERSION http://snomed.info/sct/900000000000207008 /version/20180131`
- `<< 64572001 |Disease| [{synonym = "*heart*"}] VERSION http://snomed.info/sct /900000000000207008/version/20180131`
- `<< 64572001 |Disease| [{FSN = "*heart*"}] VERSION http://snomed.info/sct/900000000000207008/version/20180131`
- `<< 64572001 |Disease| [{FSN = "*heart*"}] VERSION http://snomed.info/sct/900000000000207008/version/20180131`, LANGUAGE W
- `<< 64572001 |Disease| [{preferredTerm = "*heart*"}] VERSION http://snomed.info/sct /900000000000207008/version/20180131, LANGUAGE Y`
- `<< 64572001 |Disease| [{acceptableTerm = "*heart*"}] VERSION http://snomed.info/sct /900000000000207008/version/20180131, LANGUAGE Y`
- `("{term = "*heart*"}") VERSION http://snomed.info/sct/900000000000207008/version/20180131, LANGUAGE W`
- `X MINUS Y WHERE X = *, Y = ("{term = "*heart*"}) VERSION http://snomed.info/sct /900000000000207008/version/20180131, 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, LANGUAGE Y`
- `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|`
- `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
  - i.e. can't use a variable in its own definition
  - i.e. X is only known on the left of the corresponding 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.caseSignificanceId**
  - D.caseSignificanceId = 900000000000448009 [entire term case insensitive]
  - D.caseSignificanceId = 900000000000170005 [entire term case sensitive]
  - D.caseSignificanceId = 900000000000200002 [only initial character case insensitive]

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

- **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"
    - D.term = "*heart", D.typeId = 900000000000003001 [fully specified name]
  - FSN = "*heart" LANGUAGE X
    - D.term = "*heart", D.type = "FSN", D.acceptability = "LANGUAGE X"
    - D.term = "*heart", D.typeId = 900000000000003001 [fully specified name], acceptabilityId = "LANGUAGE X"

- **synonym**
  - synonym = "*heart"
    - D.term = "*heart", D.type = "synonym"
    - D.term = "*heart", D.typeId = 90000000000013009 [synonym]
  - synonym = "*heart" LANGUAGE X
    - D.term = "*heart", D.type = "synonym", D.acceptability = "LANGUAGE X"
    - D.term = "*heart", D.typeId = 90000000000013009 [synonym], (D.acceptabilityId = 900000000000549004 [acceptable] OR D.acceptabilityId = 900000000000548007 [preferred]) LANGUAGE X

- **synonymOrFSN**
  - synonymOrFSN = "*heart"
    - synonym = "*heart" OR FSN = "*heart"
    - D.term = "*heart", (D.type = "synonym" OR D.type = "fullySpecifiedName")
  - synonymOrFSN = "*heart" LANGUAGE X
    - synonym = "*heart" OR FSN = "*heart"
    - D.term = "*heart", (D.type = "synonym" OR D.type = "fullySpecifiedName"), D.acceptability = "LANGUAGE X"

- **textDefinition**
  - textDefinition = "*heart"
    - textDefinition = "*heart", D.type = "definition"
    - textDefinition = "*heart", D.typeId = 900000000000550004 [definition]
  - textDefinition = "*heart" LANGUAGE X
    - D.term = "*heart", D.typeId = 900000000000550004 [definition], D.acceptabilityId = "LANGUAGE X"

- **Unacceptable Terms**
  - (D.term = "*heart") MINUS (D.term = "*heart", D.acceptability = "LANGUAGE X")
### 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
- 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]

### Other topics

<table>
<thead>
<tr>
<th>Other topics</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Any other topics?</td>
</tr>
</tbody>
</table>

### Confirm next meeting date/time

<table>
<thead>
<tr>
<th>Confirm next meeting date/time</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The next SLPG meeting will be held in 6 weeks at 20:00 UTC on <strong>Wednesday 8th May</strong>.</td>
</tr>
</tbody>
</table>

**File**

No files shared here yet.