2019-05-08 - SLPG Meeting

**Date & Time**
20:00 UTC Wednesday 8th May 2019

**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**
- Make a decision on meeting at October business meeting
- Review actions from last meeting
- Proposed enhancements to template language
- Proposed enhancements to expression constraint language
  - Term searching
- Proposed new language features for mapping

**Attendees**
- Chair: Linda Bird
- Project Group: Ed Cheetham, Michael Lawley, Anne Randorff Højen, Daniel Karlsson

**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>• Question - Should we hold a SLPG meeting in October (in Malaysia)?</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>- Update template syntax with new features</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>2 new proposals to meet SNOMED International’s use cases</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1. <strong>Require slotnames to be unique</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Why? To avoid having 2 ways to do the same thing, and to clarify</td>
</tr>
<tr>
<td></td>
<td></td>
<td>the meaning through value restrictions.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Template</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[(+id)];</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1..* @morphology] = [+id @morphology ],</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[116676008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>363698007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[116676008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>363698007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Valid Expression</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Definition of 16027391000119109</td>
</tr>
<tr>
<td></td>
<td></td>
<td>64572001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[116676008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>363698007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[116676008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>363698007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>2. <strong>Introduce a sameValue constraint (similar to the allOrNone constraint)</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Template</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>[(+id)];</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[1..* @group sameValue ($morphology)]])</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Finding site] = [(+id) ,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[Associated morphology] = [(+id @morphology ) ] )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <strong>Valid Expression</strong></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(Definition of 31580001000004106</td>
</tr>
<tr>
<td></td>
<td></td>
<td>702561005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( 363698007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sacrum (body structure) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>116676008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(morphologic abnormality) ) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( 363698007</td>
</tr>
<tr>
<td></td>
<td></td>
<td>sacrum (body structure) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td>116676008</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(morphologic abnormality) ) )</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Use cases</strong>: New concept development, querying based on template</td>
</tr>
<tr>
<td></td>
<td></td>
<td>matching, and template-based modeling transformation</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Other requirements</strong></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| ]
       { |Finding site| = [ |id| (<< |Body structure| MINUS |findingSite1|) |findingSite1| ]
       |Associated morphology| = [ |id| ]
       }
       ```
     - **Valid Expression** (Definition of 16027391000119109 |Bone cyst of bilateral tibias (disorder)|)
       ```
       64572001 |Disease|:
       { |findingSite1| = [ |id| (<< |Bone structure| MINUS |findingSite1|) ]
       |Associated morphology| = [ |id| ]
       }
       ```
   - **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**
       ```
       [ |id| ]
       |Associated morphology| = [ |id| ]
       }
       ```
     - **Valid Expression** (Definition of 31580001000004106 |Bilateral sacral insufficiency fracture (disorder)|)
       ```
       702561005 |Insufficiency fracture (disorder)|:
       { |findingSite1| = [ |id| (<< |Bone structure of left half of sacrum (body structure)|) ]
       |Associated morphology (attribute)| = [ |id| ]
       }
       ```

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|) ]
       }
       ```
   - **Example B** - Default finding site of 72673000 |Bone structure|
     - **Template**
       ```
       [ |id| ]
       { |Finding site| = [ |id| ]
       }
       ```

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**
       ```
       [ |id| ]
       { |Finding site| = [ |id| ]
       }
       ```
   - **Valid Expression**
     - 195967001 |Asthma (disorder)|

4. Definition status of a templated expression
   - Specifying the definition status of a templated expression
     - **Template**
       ```
       [ |id| ]
       { |Finding site| = [ |id| ]
       }
       ```
     - **Valid Expression**
       ```
       128272009 |Disorder of lower respiratory system|:
       { |Finding site| = [ |id| ]
       }
       ```

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

**Template** - using allOrNone

- \([[[+id]]; [[[1..* @group1 allOrNone ($site), allOrNone($occurrence)]]
  [[[1..1]]] [associated morphology] = [[ +id @morphology ]],
  [[[0..1 @site]]] [Finding site] = [[ +id ]],
  [[[0..1 @occurrence]]] [Occurrence] = [[ +id ]])

**Valid Expression** - Injury of head, neck and chest

- \([[[\text{Disease}]]):
  - \([\text{Associated morphology} = \text{Injury}, \text{Finding site} = \text{Head structure}]\)
  - \([\text{Associated morphology} = \text{Injury}, \text{Finding site} = \text{Neck structure}]\)
  - \([\text{Associated morphology} = \text{Injury}, \text{Finding site} = \text{Chest structure}]\)

**Valid Expression** - Congenital malformation of head and neck

- \([[[\text{Disease}]]):
  - \([\text{Associated morphology} = \text{malformation}, \text{Finding site} = \text{Head structure},
  \text{Occurrence} = \text{Congenital}]\)
  - \([\text{Associated morphology} = \text{malformation}, \text{Finding site} = \text{Neck structure},
  \text{Occurrence} = \text{Congenital}]\)
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: Closure - 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
    - Has intent: Guidance intent
    - Using device: Core biopsy needle, device

- Valid Expression - Toilet and suture of wound
  - Procedure:
    - Method: Surgical toilet - action
    - Direct morphology: Wound
    - Procedure site - Direct: Skin structure
    - Using device: Surgical suture, 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
        - Occurrence: Genetic
        - Subject relationship context: Subject of record
      - Disease:
        - Associated morphology: Injury
        - Finding site: Neck structure
        - Occurrence: Genetic
        - Subject relationship context: Subject of record

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: Neck structure
        - Occurrence: Genetic
        - Subject relationship context: Subject of record
      - Disease:
        - Associated morphology: Injury
        - Finding site: Head structure
        - Occurrence: Genetic
        - Subject relationship context: Subject of record
Introducing term searching to ECL

**Syntax**

```plaintext
term = [lexicalSearchType ""] String
```

**Lexical Search Type**

1. Wild Card Match (collation) - e.g.
   ```plaintext
   { term = wild:"*heart*" }
   { term = wild (sv):"*hjärta*" }
   ``
2. Perl Regex - e.g.
   ```plaintext
   { term = regex:".*heart.*" }
   ``
   * Question: Does the query suddenly become order dependent? Yes
3. Word Prefix Any Order - e.g.
   ```plaintext
   { term = match:"hear att" }
   ``
   * Default (word prefix any order) - e.g.
   ```plaintext
   { term = "hear att" }
   { term = "*heart*" }
   ```

**Potential Examples**

- `<64572001|Disease|{{ term = "*heart*" }}>
- `<64572001|Disease|{{ term = "*heart*", languageCode = "en" }}>
- `<64572001|Disease|{{ term = "*cardio*" }}>
- `<64572001|Disease|{{ term = "*heart*", languageCode = "sv" }}>
- `<64572001|Disease|{{ term = "*cardio*" }}>

**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| }}>
- Do we include 'type' - e.g. `<64572001|Disease|{{ D.term = "*heart*", D.type = synonym }}>
- Do we include 'languageCode' - e.g. `<64572001|Disease|{{ D.term = "*heart*", D.type = synonym, D.languageCode = "en" }}>
- Do we include 'caseSignificanceId' - e.g. `<64572001|Disease|{{ D.term = "Heart", D.caseSignificanceId = 900000000000017005 |case sensitive| }}>
- Do we include 'caseSignificance' - e.g. `<64572001|Disease|{{ D.term = "*Heart*", D.caseSignificance = sensitive }}>
- Do we include 'language' and 'version' - e.g. `<64572001|Disease|{{ D.term = "*heart*" }} VERSION = http://...., LANGUAGE = (999001881000000108|Gastro LRS|, |GB English|)}
- Do we include syntactic sugar - e.g.
  ```plaintext
  { preferredTerm = "*heart*", languageRefSet = en-gb } FROM version = X, language = Y
  ```
- Do we use/require the "D" at the start of "term"?
- How does this relate to description templates? How does this relate to SNOSTORM implementation.
- Packaging - Do we (over time) extend ECL with all the new Query Language features, and define a set of example ECL profiles - e.g. "Basic ECL", "ECL with basic term searching", "ECL with filters"
Proposed 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 + Attribute refinement**
    - `|Anatomy structure and part association refset| . |mapTarget|`
    - `|Anatomy structure and part association refset| . |referencedComponent|`
    - (Same as `|Anatomy structure and part association refset|: |referencedComponent| = << |Upper abdomen structure|)
  - **Dot notation + Filters**
    - `( |Anatomy structure and part association refset| {{ |referencedComponent| = << |Upper abdomen structure| }} ). |mapTarget|`
    - `( |Anatomy structure and part association refset| {{ |mapTarget| = << |Upper abdomen structure| }} ) . |referencedComponent|`
  - **Specify value to be returned**
    - `?|mapTarget|? ( |Anatomy structure and part association refset|)`
    - `?|mapTarget|? ( |Anatomy structure and part association refset|: |referencedComponent| = << |Upper abdomen structure|)`

Proposal from Michael:

- 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|)`

- **Finalize and publish language and language instance URIs**
<table>
<thead>
<tr>
<th>Query</th>
<th>Language</th>
<th>Linda Bird</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language Summary from previous meetings</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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/ve rsion/20180131`
- `<< 64572001 |Disease| {{ FSN = "*heart*" }} VERSION http://snomed.info/sct/900000000000207008/ver sion/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`
- `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 X WHERE X = (<< 1234 : 5678 = 6547), Y = (<< 1456) 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, LANGUAGE 900000000000508004 [GB English]`
- `X MINUS X WHERE X = (<< 1234 : 5678 = 6547), Y = (<< 1456), LANGUAGE 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
  - X can’t use a variable in its own definition
  - 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 = wildcard: "*heart*"
  - D.term = regex: ".*heart.*"
  - D.term = matches: "hear att"
  - D.term = (sv) wildcard: "*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"
    - D.term = "*heart", D.typeId = 900000000000003001 [fully specified name]

- **synonym**
  - synonym = "*heart"
    - D.term = "*heart", D.type = "synonym"
    - D.term = "*heart", D.typeId = 900000000000013009 [synonym]

- **textDefinition**
  - textDefinition = "*heart"
    - D.term = "*heart", D.typeId = 900000000000550004 [definition]

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]

<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>

<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 22nd May</strong>.</td>
<td></td>
</tr>
</tbody>
</table>