

# 2019-01-30 - SLPG Meeting

## Date & Time

20:00 UTC Wednesday 30th January 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

- Consider proposal to change language syntax from ABNF to ANTLR
- Recap ECL transitivity/role chain proposal, and consider implementation support
- Summarize 2019 work items, including:
  - Proposed new ECL language features
  - Updates to URI standard
  - Enhancement to template language
  - Draft Query Language

## Attendees

- Chair: [Linda Bird](#)
- Project Group: [Daniel Karlsson](#), [Michael Lawley](#), [Anne Randorff Højten](#), [Kai Kewley](#), [Ed Cheetham](#), [Rob Hausam](#), [Guillermo Reynoso](#), [Harold Solbrig](#)

## Apologies

## Agenda and Meeting Notes

Description	Owner	Notes
Welcome and apologies	<a href="#">Linda Bird</a>	
Proposal to move from ABNF to ANTLR	<a href="#">Daniel Karlsson</a>	<p>For discussion: Daniel has proposed to change the primary syntax representation for all languages from ABNF to ANTLR.</p> <ul style="list-style-type: none"><li>• Auto-translation from ABNF to ANTLR difficult</li><li>• ABNF grammar represents the byte representation of UTF-8 / Needs to be updated to include a range of UTF characters (not the byte representation)</li><li>• ABNF is a tool-independent language for defining syntaxes / ANTLR is a tool for defining parsers (and dependent on versions)</li><li>• Proposal - Continue to use ABNF as the standard, with a hand-crafted ANTLR as the authoritative ANTLR reference implementation</li><li>• Alternate representations, e.g. JSON, would be useful (e.g. for templates)</li><li>• Actions<ul style="list-style-type: none"><li>◦ Post authoritative ANTLR syntax used by SNOMED International</li><li>◦ Update ABNF with additional UTF characters</li></ul></li></ul>
Transitivity, Reflexivity & Role chaining	<a href="#">Linda Bird</a> <a href="#">Kai Kewley</a>	<p>Proposed extension to ECL to support transitive relationships and role chaining (to align with new enhanced DL axioms)</p> <ul style="list-style-type: none"><li>• Example 1<ul style="list-style-type: none"><li><b>Direct relationship</b><ul style="list-style-type: none"><li>• &lt;  Body structure : &lt;&lt; 774081006  Proper part of  = &lt;&lt; 51185008  Chest </li></ul></li><li><b>Transitive relationship</b><ul style="list-style-type: none"><li>• &lt;  Body structure : &lt;&lt; 774081006  Proper part of * = &lt;&lt; 51185008  Chest </li><li>• &lt;&lt;(&lt;  Body structure : &lt;&lt; 774081006  Proper part of  = &lt;&lt; 51185008  Chest )</li></ul></li></ul></li><li>• Example 2<ul style="list-style-type: none"><li><b>Direct relationship</b><ul style="list-style-type: none"><li>• &lt; 71388002   : 363701004  Direct substance  = 372687004  Amoxicillin </li></ul></li><li><b>Role chained relationship (via 738774007 [is modification of])</b><ul style="list-style-type: none"><li>• &lt; 71388002   : 363701004  Direct substance * = 372687004  Amoxicillin </li><li>• &lt;&lt;(&lt; 71388002   : 363701004  Direct substance  = 372687004  Amoxicillin )</li></ul></li></ul></li></ul> <p>What implementation support will be required? Should we provide easy access to those relationships that can be inferred by transitivity and role chains (note: These will be excluded from the inferred relationship file as they are redundant). If so, then what format should be used - for example, a TSV file with the following columns:</p> <ul style="list-style-type: none"><li>◦ sourceId</li><li>◦ destinationId</li><li>◦ typeId</li><li>◦ relationshipGroup</li></ul>
END OF MEETING		

Executing maps	Linda Bird	<p>Proposed extension to ECL to support the execution of maps (focusing on the resolution of historical refsets)</p> <ul style="list-style-type: none"> <li>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 that identifies a set of concepts 'c' to be used for retrieving patient records, you probably also want to retrieve records for sameAs (c) and replacedWith (c) <ul style="list-style-type: none"> <li>Example: <ul style="list-style-type: none"> <li>(&lt; 72704001  Fracture  AND ^ 900000000000527005  SAME AS association reference set ) . 9000 00000000533001  Association target component </li> </ul> </li> </ul> </li> </ul>
Template Syntax	Linda Bird	<p>New requirements</p> <ul style="list-style-type: none"> <li>2 slots must have the same value</li> <li>2 slots must have different values</li> <li>The value of 1 slot must subsume the value of another</li> <li>Default value for slots</li> </ul>
URI Standard	Linda Bird	<ul style="list-style-type: none"> <li>Finalize and publish language and language instance URIs</li> </ul>
Query Language - Summary from previous meetings	Linda Bird	<p><b>Examples: version and language</b></p> <ul style="list-style-type: none"> <li>&lt;&lt; 64572001  Disease  {{ term = "heart*" }} <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></li> <li>&lt;&lt; 64572001  Disease  {{ synonym = "heart*" }} <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></li> <li>&lt;&lt; 64572001  Disease  {{ FSN = "heart*" }} <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></li> <li>&lt;&lt; 64572001  Disease  {{ FSN = "heart*" }} <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, <b>LANGUAGE</b> W</li> <li>&lt;&lt; 64572001  Disease  {{ preferredTerm = "heart*" }} <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, <b>LANGUAGE</b> Y</li> <li>&lt;&lt; 64572001  Disease  {{ acceptableTerm = "heart*" }} <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, <b>LANGUAGE</b> Y</li> <li>(* {{ term = "heart*" }} <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, <b>LANGUAGE</b> Z) MINUS (* {{ term = "heart*" }} <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20170731">http://snomed.info/sct/900000000000207008/version/20170731</a>, <b>LANGUAGE</b> W)</li> <li>X MINUS Y <b>WHERE</b> X = * , Y = (* {{ term = "heart*" }}) <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, <b>LANGUAGE</b> W</li> </ul> <p><b>Notes</b></p> <ul style="list-style-type: none"> <li>Allow nested where, version, language</li> <li>Scope of variables is inner query</li> </ul>
		<p><b>Examples: where</b></p> <ul style="list-style-type: none"> <li>X MINUS &gt;! X <b>WHERE</b> X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547)</li> <li>X MINUS &gt;! X <b>WHERE</b> X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547) <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></li> <li>X MINUS &gt;! Y <b>WHERE</b> X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547), Y = (&lt;&lt; 1456) <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a></li> <li>X MINUS &gt;! X <b>WHERE</b> X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547) <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, <b>LANGUAGE</b> 900000000000508004  GB English </li> <li>X MINUS &gt;! X <b>WHERE</b> X = (&lt;&lt; 1234 : 5678 = &lt;&lt; 6547) <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, <b>LANGUAGE</b> 999001881000000108 GB clinical extension LRS , 900000000000508004  GB English </li> <li>X minus &gt;! X <b>WHERE</b> X = ( &lt; M <b>WHERE</b> M = (&lt; 1234))) <b>VERSION</b> <a href="http://snomed.info/sct/900000000000207008/version/20180131">http://snomed.info/sct/900000000000207008/version/20180131</a>, <b>LANGUAGE</b> 999001881000000108 GB clinical extension LRS , 900000000000508004  GB English </li> </ul> <p><b>Notes</b></p> <ul style="list-style-type: none"> <li>Allow nested variable definitions, but recommend that people don't due to readability</li> <li>Scope of variables is the inner query</li> <li>No recursion e.g X <b>WHERE</b> X = 1234 MINUS X <ul style="list-style-type: none"> <li>ie can't use a variable in its own definition</li> <li>ie X is only known on the left of the corresponding <b>WHERE</b>, and not on the right of the <b>WHERE</b></li> </ul> </li> </ul>

#### 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.caseSignificancelId**
  - D.caseSignificancelId = 900000000000448009 [entire term case insensitive]
  - D.caseSignificancelId = 900000000000017005 [entire term case sensitive]
  - D.caseSignificancelId = 900000000000020002 [only initial character case insensitive]
- **D.caseSignificance**
  - D.caseSignificance = `"insensitive"`
  - D.caseSignificance = `"sensitive"`
  - D.caseSignificance = `"initialCharInsensitive"`
- **D.typeId**
  - D.typeId = 90000000000003001 [fully specified name]
  - D.typeId = 900000000000013009 [synonym]
  - D.typeId = 900000000000055004 [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 = 90000000000003001 [fully specified name]
  - FSN = `"*heart" LANGUAGE X`
    - D.term = `"*heart"`, D.type = `"FSN"`, D.acceptability = `* LANGUAGE X`
    - D.term = `"*heart"`, D.typeId = 90000000000003001 [fully specified name], acceptabilityId = `* LANGUAGE X`
- **synonym**
  - synonym = `"*heart"`
    - D.term = `"*heart"`, D.type = `"synonym"`
    - D.term = `"*heart"`, D.typeId = 900000000000013009 [synonym]
  - synonym = `"*heart" LANGUAGE X`
    - D.term = `"*heart"`, D.type = `"synonym"`, D.acceptability = `* LANGUAGE X`
    - D.term = `"*heart"`, D.typeId = 900000000000013009 [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" LANGUAGE X`
    - D.term = `"*heart"`, (D.type = `"synonym"` OR D.type = `"fullySpecifiedName"`), D.acceptability = `* LANGUAGE X`
- **textDefinition**
  - textDefinition = `"*heart"`
    - D.term = `"*heart"`, D.type = `"definition"`
    - D.term = `"*heart"`, D.typeId = 900000000000055004 [definition]
  - textDefinition = `"*heart" LANGUAGE X`
    - D.term = `"*heart"`, D.type = `"definition"`, D.acceptability = `* LANGUAGE X`
    - D.term = `"*heart"`, D.typeId = 900000000000055004 [definition], D.acceptabilityId = `* LANGUAGE X`
- **Unacceptable Terms**
  - (D.term = `"*heart"`) MINUS (D.term = `"*heart"`, D.acceptability = `* LANGUAGE X`)

		<p>Language preferences using multiple language reference sets</p> <ul style="list-style-type: none"> <li>• 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 <ul style="list-style-type: none"> <li>◦ E.g. Regional LRS that adds local dialect to a National LRS</li> <li>◦ E.g. Specialty-specific LRS</li> <li>◦ E.g. Irish LRS that adds local preferences to the en-GB LRS <ul style="list-style-type: none"> <li>▪ 99999900  Irish language reference set  <b>PLUS</b>  GB English reference set </li> </ul> </li> </ul> </li> <li>• 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 <ul style="list-style-type: none"> <li>◦ E.g. Danish LRS that does a partial translation of the International Release <ul style="list-style-type: none"> <li>▪ 999999  Danish language reference set  <b>ELSE</b>  GB English reference set </li> </ul> </li> </ul> </li> </ul>
Other topics	<a href="#">Linda Bird</a>	<ul style="list-style-type: none"> <li>• Any other topics?</li> </ul>
Confirm next meeting date /time	<a href="#">Linda Bird</a>	The next SLPG meeting will be held in 2 weeks at 20:00 UTC on <b>Wednesday 6th February</b> .

**File    Modified**

No files shared here yet.