

2020-03-11 - SLPG Meeting

Date & Time

20:00 UTC Wednesday 11th March 2020

Goals

- To finalize URI updates for publication
- To finalize requirements for term searching in ECL

Location

Zoom meeting: <https://snomed.zoom.us/j/471420169>

Attendees

- Chair: [Linda Bird](#)
- Project Group: [Michael Lawley](#), [Daniel Karlsson](#), [Peter Jordan](#), [Rob Hausam](#), [Ed Cheetham](#)

Apologies

Agenda and Meeting Notes

Description	Owner	Notes
Welcome and agenda	Linda Bird	SNOMED International has announced the cancellation of the face to face April 2020 business meetings. Therefore, the SNOMED Languages meeting on Sunday 5th April is also cancelled.
Concrete values	Linda Bird	ON HOLD: SCG, ECL, STS, ETL - Ready for publication, but on hold until after MAG meeting in April confirming requirement for Boolean datatype.
URIs	Linda Bird	DISCUSSION: Do we have any use cases for URIs for language syntaxes and instances? TO BE DECIDED: Which new URI format(s) will be included in the revisions to the URI standard? Draft URI standard for review - URI Standard <ul style="list-style-type: none">• 2.6 URIs for Language Syntaxes - Not for publication• 2.7 URIs for Language Instances - Not for publication• 2.8 URIs for Modelling Resources * - To publish• 3.1 Resolving SNOMED CT URIs * - To publish (with revised wording)
Expression Constraint Language	Linda Bird	FUTURE PLANS <ul style="list-style-type: none">• Feature Request - Child or self (<<!) has been requested by an implementation in U.S.<ul style="list-style-type: none">◦ Use case: Connect our medical content (medical articles and doctor specialties/treatments) on a hospital's website. They do not need high granularity for the approx 10,000 concepts needed to choose to tag an article or use on a doctor's profile page. Most often the parent concept, or one of its direct children is all that is needed. They are creating a refset of only the concepts we may need for tagging content. The descendants operators provided too many results. In most cases, the parent and its direct children will work well for their needs. While the "<! X OR X" works, they would prefer to eliminate keystrokes or ECL characters.◦ >>! for ancestor or self (>>!)• Term Searching: Agreement in Malaysia - ECL will add the following term searching syntax (no regex - just wild card and word prefix any order): <pre>{{ term = [termSearchType :] "String", language = <langCode> }}</pre>• Example - {{ term = "heart att", language = es }}• What other optional parameters should be included? - e.g. 'type', 'dialect', 'acceptability'<ul style="list-style-type: none">◦ typeId = 90000000000000013009 ; type = <syn fsn textdef>◦ dialectId = 9000000000000508004 [GB English] ; dialect = <en-GB en-AU nz-Patient en-Patient de-CardioSpecialist><ul style="list-style-type: none">▪ dialectId = 9000000000000508004 + 9000000000000509007 ; dialect = en-GB + en-US▪ Dialect_Alias refset : alias + languageRefset-conceptId - e.g. "en-GB", 9000000000000508004◦ acceptabilityId = 9000000000000549004 ; acceptability = <acceptable preferred >• Use case - Searching on PT Term Search Type

- a. Wild Card Match (collation) - e.g.
 - o {{ term = **wild**:**heart** }}
 - o {{ term = wild (sv):**hjärta** }}
- a. Word Prefix Any Order - e.g.
 - o {{ term = **match**:hear att }}
- a. Default (word prefix any order) - e.g.
 - o {{ term = "hear att" }}
 - o {{ term = "**heart**" }}

Potential Examples

- o << 64572001 |Disease| {{ term = "heart" }}
- o << 64572001 |Disease| {{ term = "heart", language = "en" }}
- o << 64572001 |Disease| {{ term = "heart", language = "en" }} AND << 64572001 |Disease| {{ term = "hjärta", language = "sv" }}
- o << 64572001 |Disease| {{ term = "heart", language = "en" }} {{ term = "hjärta", language = "sv" }}
- o << 64572001 |Disease| {{ term = "heart", language = "en" }} OR << 64572001 |Disease| {{ term = "hjärta", language = "sv" }}
- o << 64572001 |Disease| {{ (term = "heart", language = "en") OR (term = "hjärta", language = "sv") }}
- o (<< 64572001 |Disease|: |Associated morphology| = *) {{ term = "heart", language = "en", }} {{ term = "hjärta", language = "sv" }}
- o (<< 64572001 |Disease| {{ term = "**cardio**" }}) MINUS (<< 64572001 |Disease| {{ term != "**heart**" }})
- o Recommendation to be made on (based on investigation of grammar):
 - << 64572001 |Disease| {{ term = "heart", language = "en" }} AND {{ term = "hjärta", language = "sv" }}
 - << 64572001 |Disease| ({{ term = "heart", language = "en" }} OR {{ term = "hjärta", language = "sv" }})
 - << 64572001 |Disease| ({{ term = "heart", language = "en" }} MINUS {{ term = "hjärta", language = "sv" }})

Use Cases

- o 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.
- o Authors quality assuring names of concepts
- o Checking translations, retranslating. Queries for a concept that has one word in Swedish, another word in English
- o AU use case would have at most 3 or 4 words in match
- o Consistency of implementation in different terminology services
- o Authoring use cases currently supported by description templates
- o A set of the "ectomy"s and "itis"s

Questions

- o Do we include 'typeld' - e.g. << 64572001 |Disease| {{ D.term = "**heart**", typeld = 900000000000013009 |Synonym| }}
 - NO
- o Do we include 'type' - e.g. << 64572001 |Disease| {{ D.term = "**heart**", D.type = synonym }}
 - NO
- o Do we include 'languageCode' - e.g. << 64572001 |Disease| {{ D.term = "**heart**", D.type = synonym, D.languageCode = "en" }}
 - YES
- o Do we include 'caseSignificancelld' - e.g. << 64572001 |Disease| {{ D.term = "**Heart**", D.caseSignificancelld = 900000000000017005 |case sensitive| }}
 - NO
- o Do we include 'caseSignificance' - e.g. << 64572001 |Disease| {{ D.term = "**Heart**", D.caseSignificance = sensitive }}
 - NO
- o Do we include 'language' and 'version' - e.g. << 64572001 |Disease| {{ term = "**heart**" }} VERSION = http://..., LANGUAGE = (999001881000000108|Gastro LRS|, |GB English|)
 - NO
- o Do we include syntactic sugar - e.g.
 - << 64572001 |Disease| {{ preferredTerm = "**heart**", languageRefSet = en-gb }}
 - << 64572001 |Disease| {{ fullySpecifiedTerm = "**heart**", languageRefSet=en-gb }}
 - << 64572001 |Disease| {{ acceptableTerm = "**heart**", languageRefSet = en-gb }}
 - << 64572001 |Disease| {{ preferredTerm = "**heart**" }} FROM version = X, language = Y
 - NO
- o Do we use/require the "D" at the start of "term"?
 - NO
- o Packaging - How do we package this extension to ECL
 - A new version of ECL - version 1.5

Querying
Refset

Linda
Bird

Proposed syntax to support querying and return of alternative refset attributes (To be included in the SNOMED Query Language)

Attributes

- **Example use cases**

- Execution of maps from international substance concepts to AMT substance concepts
- Find the anatomical parts of a given anatomy structure concept (in |Anatomy structure and part association reference set)
- Find potential replacement concepts for an inactive concept in record
- Find the order of a given concept in an Ordered component reference set
- Find a concept with a given order in an Ordered component reference set

- Potential syntax to consider (brainstorming ideas)
 - **SELECT ??**
 - SELECT 123 |referenced component|, 456 |target component|
FROM 799 |Anatomy structure and part association refset|
WHERE 123 |referenced component| = (< 888 |Upper abdomen structure| {{ term = "heart" }})
 - SELECT id, moduleId
FROM concept
WHERE id IN (< |Clinical finding|)
AND definitionStatus = |primitive|
 - SELECT id, moduleId
FROM concept, ECL("< |Clinical finding") CF
WHERE concept.id = CF.sctid
AND definitionStatus = |primitive|
 - SELECT ??? |id|, ??? |moduleId|
FROM concept (< |Clinical finding| {{ term = "heart" }} {{ definitionStatus = |primitive| }})
 - **Question** - Can we assume some table joins - e.g. Concept.id = Description.conceptId etc ??
 - **Examples**
 - Try to recast relationships table as a Refset table + graph-based extension
 - Find primitive concepts in a hierarchy
 - **ROW ... ?**
 - ROWOF (|Anatomy structure and part association refset|) ? (|referenced component| , |target component|)
 - same as: ^ |Anatomy structure and part association refset|
 - ROWOF (|Anatomy structure and part association refset|) . |referenced component|
 - same as: ^ |Anatomy structure and part association refset|
 - ROWOF (|Anatomy structure and part association refset|) {{ |referenced component| = << |Upper abdomen structure| }} ? |targetComponentId|
 - ROWOF (< 900000000000496009|Simple map type reference set| {{ term = "My hospital" }}) {{ 449608002|Referenced component| = 80581009 |Upper abdomen structure| }} ?
900000000000505001 |Map target|
 - (ROW (< 900000000000496009|Simple map type reference set| {{ term = "My hospital" }}) : 449608002|Referenced component| = 80581009 |Upper abdomen structure|). 900000000000505001 |Map target|
 - **# ... ?**
 - # |Anatomy structure and part association refset| ? |referenced component|
 - # (|Anatomy structure and part association refset| {{|referenced component| = << |Upper abdomen structure| }} ? |targetComponentId|
 - **? notation + Filter refinement**
 - |Anatomy structure and part association refset| ? |targetComponentId|
 - |Anatomy structure and part association refset| ? |referencedComponent| (Same as ^ |Anatomy structure and part association refset|)
(|Anatomy structure and part association refset| {{ |referencedComponent| = << |Upper abdomen structure| }} ? |targetComponentId|
 - (|Anatomy structure and part association refset| {{ |targetComponentId| = << |Upper abdomen structure| }}) ? |referencedComponent|
 - (|My ordered component refset| : |Referenced component| = |Upper abdomen structure|) ? **|priority order|**
 - ? |My ordered component refset| {{ |Referenced component| = |Upper abdomen structure| }}
. |priority order|
 - ? |My ordered component refset| . |referenced component|
 - equivalent to ^ |My ordered component refset|
 - ? (<|My ordered component refset|) {{ |Referenced component| = |Upper abdomen structure| }} . |priority order|
 - ? (<|My ordered component refset| {{ term = "map" }}) {{ |Referenced component| = |Upper abdomen structure| }} . |priority order|
 - REFSETROWS (<|My ordered component refset| {{ term = "map" }}) {{ |Referenced component| = |Upper abdomen structure| }} SELECT |priority order|
 - **Specify value to be returned**
 - ? 449608002 |Referenced component|?
734139008 |Anatomy structure and part association refset|
 - ^ 734139008 |Anatomy structure and part association refset| (Same as previous)
 - ? 900000000000533001 |Association target component|?
734139008 |Anatomy structure and part association refset|
 - ? 900000000000533001 |Association target component|?
734139008 |Anatomy structure and part association refset| :
449608002 |ReferencedComponent| = << |Upper abdomen structure|
 - ? 900000000000533001 |Association target component|?
734139008 |Anatomy structure and part association refset|
{{ 449608002 |referencedComponent| = << |Upper abdomen structure| }}
 - (? 900000000000533001 |Association target component|?
734139008 |Anatomy structure and part association refset| :
449608002 |ReferencedComponent| = (<< |Upper abdomen structure|) : |Finding site| = *)

Returning Attributes	Michael Lawley	<p>Proposal (by Michael) for discussion</p> <ul style="list-style-type: none"> 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. <p>For example, I can write:</p> <pre><< 404684003 Clinical finding : 363698007 Finding site = <<66019005 Limb structure </pre> <pre><< 404684003 Clinical finding . 363698007 Finding site </pre> <p>But I can't get all the attribute names that are used by << 404684003 Clinical finding </p> <ul style="list-style-type: none"> Perhaps something like: <ul style="list-style-type: none"> ? R.type ? (<< 404684003 Clinical finding) This could be extended to, for example, return different values - e.g. <ul style="list-style-type: none"> ? Simple map refset . maptarget ? (^ Simple map refset AND < Fracture)
Reverse Member Of	Michael Lawley	<p>Proposal for discussion</p> <p>What refsets is a given concept (e.g. 421235005 Structure of femur) a member of?</p> <ul style="list-style-type: none"> Possible new notation for this: <ul style="list-style-type: none"> ^ . 421235005 Structure of femur ? X ? 421235005 Structure of femur = ^ X

Expression Templates	Peter G. Williams	<ul style="list-style-type: none"> • ON HOLD WAITING FROM IMPLEMENTATION FEEDBACK FROM INTERNAL TECH TEAM • WIP version - https://confluence.ihtsdotools.org/display/WIPSTS/Template+Syntax+Specification <ul style="list-style-type: none"> ▪ Added a 'default' constraint to each replacement slot - e.g. default (72673000 Bone structure (body structure)) ▪ Enabling 'slot references' to be used within the value constraint of a replacement slot - e.g. [[+id (<< 123037004 Body structure MINUS << \$findingSite2) @findingSite1]] ▪ Allowing repeating role groups to be referenced using an array - e.g. \$rolegroup[1] or \$rolegroup[!=SELF] ▪ Allow reference to 'SELF' in role group arrays ▪ Adding 'sameValue' and 'allOrNone' constraints to information slots - e.g. sameValue (\$site), allOrNone (\$occurrence) ▪ See changes in red here: 5.1. Normative Specification <p>Examples:</p> <pre>[[+id]]: [[1..*] @my_group sameValue(morphology)] { Finding site = [[+id (<<123037004 Body structure (body structure) MINUS << \$site[! SELF]) @site] , Associated morphology = [[+id @my_morphology]] }</pre> <ul style="list-style-type: none"> • Implementation feedback on draft updates to Expression Template Language syntax <ul style="list-style-type: none"> ◦ Use cases from the Quality Improvement Project: <ul style="list-style-type: none"> ▪ Multiple instances of the same role group, with some attributes the same and others different. Eg same morphology, potentially different finding sites. <p>Note that QI Project is coming from a radically different use case. Instead of <i>filling</i> template slots, we're looking at existing content and asking "exactly <i>how</i> does this concept fail to comply to this template?"</p> <p>For discussion:</p> <pre>[[0..1]] { [[0..1]] 246075003 Causative agent = [[+id (< 410607006 Organism) @Organism]] }</pre> <p>Is it correct to say either one of the cardinality blocks is redundant? What are the implications of 1..1 on either side? This is less obvious for the self grouped case.</p> <p>Road Forward for SI</p> <ol style="list-style-type: none"> 1. Generate the parser from the ABNF and implement in the Template Service 2. User Interface to a) allow users to specify template at runtime b) tabular (auto-completion) lookup STL 3. Template Service to allow multiple templates to be specified for alignment check (aligns to none-off) 4. Output must clearly indicate exactly what feature of concept caused misalignment, and what condition was not met. <p>Additional note: QI project is no longer working in subhierarchies. Every 'set' of concepts is selected via ECL. In fact most reports should now move to this way of working since a subhierarchy is the trivial case. For a given template, we additionally specify the "domain" to which it should be applied via ECL. This is much more specific than using the focus concept which is usually the PPP eg Disease.</p> <p>FYI Michael Chu</p>
Description Templates	Kai Kewley	<ul style="list-style-type: none"> • ON HOLD • Previous discussion (in Malaysia) <ul style="list-style-type: none"> ▪ Overview of current use ▪ Review of General rules for generating descriptions <ul style="list-style-type: none"> • Removing tags, words • Conditional removal of words • Automatic case significance • Generating PTs from target PTs • Reordering terms ▪ Mechanism for sharing general rules - inheritance? include? ▪ Description Templates for translation ▪ Status of planned specification

<p>Query Language - Summary from previous meetings</p>	<p>Linda Bird</p>	<div data-bbox="405 226 1445 898"> <p>FUTURE WORK</p> <p>Examples: version and dialect</p> <ul style="list-style-type: none"> ◦ << 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, DIALECT W ◦ << 64572001 Disease {{ preferredTerm = "heart*" }} VERSION http://snomed.info/sct/900000000000207008/version/20180131, DIALECT Y ◦ << 64572001 Disease {{ acceptableTerm = "heart*" }} VERSION http://snomed.info/sct/900000000000207008/version/20180131, DIALECT Y ◦ (* {{ term = "heart*" }} VERSION http://snomed.info/sct/900000000000207008/version/20180131, DIALECT Z) MINUS (* {{ term = "heart*" }} VERSION http://snomed.info/sct/900000000000207008/version/20170731, DIALECT W) ◦ X MINUS Y WHERE X = * , Y = (* {{ term = "heart*" }}) VERSION http://snomed.info/sct/900000000000207008/version/20180131, DIALECT W <p>Notes</p> <ul style="list-style-type: none"> ◦ Allow nested where, version, language ◦ Scope of variables is inner query </div> <div data-bbox="405 898 1445 1480"> <p>Examples: where</p> <ul style="list-style-type: none"> ◦ 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, DIALECT 999001881000000108 GB clinical extension LRS , 900000000000508004 GB English <p>Notes</p> <ul style="list-style-type: none"> ▪ 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 <ul style="list-style-type: none"> • 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 </div>
--	-------------------	--

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 = 900000000000050004 [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 = 900000000000050004 [definition]
 - textDefinition = `"*heart" LANGUAGE X`
 - D.term = `"*heart"`, D.type = `"definition"`, D.acceptability = `* LANGUAGE X`
 - D.term = `"*heart"`, D.typeId = 900000000000050004 [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"> • 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"> ◦ 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 <ul style="list-style-type: none"> ▪ 99999900 Irish language reference set PLUS GB English reference set • 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"> ◦ E.g. Danish LRS that does a partial translation of the International Release <ul style="list-style-type: none"> ▪ 999999 Danish language reference set ELSE GB English reference set
Confirm next meeting date /time	Linda Bird	Next meeting is scheduled for Wednesday 11th March 2020 at 20:00 UTC.

File Modified

No files shared here yet.