2020-08-26 - SLPG Meeting

Date & Time

20:00 to 21:00 UTC Wednesday 26th August 2020

Goals

- To walk through complete draft of ECL v1.5
- To discuss next steps

Location

Zoom meeting link (password: 764978)

Attendees

- Chair: Linda Bird
- Project Group: Michael Lawley, Anne Randorff Højen, Peter Jordan, Daniel Karlsson, Ed Cheetham, Rob Hausam

Apologies

Agenda and Meeting Notes

Description	Owner	Notes
Welcome and agenda	Linda Bird	**NOTE**: Next meeting to be held on Wednesday 9th September
Concrete Values	Linda Bird	**Specifications** • **SCG v2.4** (with booleans) has been published • **ECL v1.4** (with booleans, childOrSelfOf and parentOrSelfOf) has been published • **MRCM** (with updated rangeConstraint) - 5.3 MRCM Attribute Range Reference Set • **STS v1.1** and **ETL v1.1** (with booleans) to be published soon

Expression Constraint Language	Linda Bird	• **Updates to WIP ECL v1.5**
		  ○ 1. Introduction - History updated to mention ECL v1.5
		  ○ 3.2 Expression Constraint and Query Requirements - The 'term filter' requirement is explained.
		  ○ 4. Logical Model - Filters are added to the abstract model, and the example expression constraint (with marked components)
		  ○ 4.1 Details - Filter constraints added to the UML class diagram
		  ○ 5.1 Brief Syntax (Normative) - Syntax updated with filters
		  ○ 5.2 Long Syntax (Informative) - Syntax updated with filters
		  ○ 5.3 Informative Comments - Comments added for new rules
		  ○ 5.5 Character Collation for Term Filters - New section. Default collation still needs to be defined
		  ○ 6.8 Description Filters - New page with filter constraint examples
		  ○ Appendix A.7 - Valid examples with filter constraints (to be added)
		  ○ Appendix B.7 - Invalid examples with filter constraints (to be added)
		  ○ Appendix C - Dialect Aliases - Proposed dialect aliases for a range of refsets accessible through browser
		• **THIS WEEK**
		  ▪ **Please review all the new content for ECL v1.5 (above) and provide feedback**
		  ▪ **Next Steps: WIP ECL v1.5 will be reviewed internally in SNOMED International and by MAG**
		  ▪ Final questions:
		  • Dialect Alias comment from Daniel - i.e. are alpha characters sufficient?
		  • Subexpression comment from Ed - i.e. New example with refinement, and clarification text on use of MINUS
		  • Preferred Term filters on dialect lists (first refset? or both refsets?) - needs clarification
		  • Appendix A.7 and B.7 to be completed (once examples agreed upon)
		  • How are the default collation rules determined?
		  a. The 'Language' property of each description (note: descriptions can be pre-indexed based on their language)
		  b. The 'Language reference set' being used? (note: each language refset would need to be associated with a locale)
		  c. In the ECL query? Language filter? Dialect filter?
		  d. The local environment
		  e. A combination of these
		• **On Hold**
		  ○ Can/should we register ECL as a MIME type? – Waiting for volunteer time to complete registration form
		• **To be progressed through the MAG**
		  1. Dialect Alias Refset
		  Alternative 1 - Annotation Refset
		  • Dialect_Alias refset : alias + languageRefset-conceptId - e.g. "en-GB", 900000000000508004
		  Example row
		  ○ referencedComponentId = 999001261000000100
		  ○ dialectAlias = nhs-clinical
		  Alternative 2 - Add alias as a synonym to the language refset concept
		  • Create a simple type refset that refers to the preferred alias for each language refset
		  Alternative 3 - Use non-defining relationships on the language refset concept
		• **1. 2. Constructing a Language Refset from other Language Refset**
		  ▪ Allowing an intensional definition for a language refset
		  ▪ Includes order/precedence of language refsets being combined

URIs for Extended Editions		How to refer to an 'extended edition' using a URI - e.g. "International Edition plus the following 2 nursing modules: 733983009 |IHTSDO Nursing Health Issues module|and 733984003 |IHTSDO Nursing Activities module| Use Case - Need to execute an ECL, that refers to "^ 733991000 | Nursing Health Issues Reference Set (foundation metadata concept) |" and/or "^ 733990004 | Nursing Activities Reference Set (foundation metadata concept) |", where the substrate includes the international edition, plus the modules that include these reference sets July 2020 International Edition URI: http://snomed.info/sct/900000000000207008/version/20200731 July 2020 International Edition + nursing modules URI ?? - For example: • http://snomed.info/sct/900000000000207008/version/20200731/module/733983009/time/20200131/module/733984003/time/20200131 • http://snomed.info/sct/900000000000207008/version/20200731/modules/733983009:733984003 • http://snomed.info/sct/900000000000207008:733983009:733984003/version/20200731:20190731:20200131 • Canonical order? Or order doesn't matter? • Constraints on what can go in the additional packages (only refsets and their metadata)

Querying Refset Attributes	Linda Bird	Proposed syntax to support querying and return of alternative refset attributes (To be included in the SNOMED Query Language)
		
		• **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 struture 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	**Proposal (by Michael) for discussion** • 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|)
Reverse Member Of	Michael Lawley	**Proposal for discussion** What refsets is a given concept (e.g. 421235005 |Structure of femur|) a member of? • Possible new notation for this: ○ ^ . 421235005 |Structure of femur| ○ ? X ? 421235005 |Structure of femur| = ^ X

Expression Templates	Peter G. Williams	• **ON HOLD WAITING FROM IMPLEMENTATION FEEDBACK FROM INTERNAL TECH TEAM**
		• WIP version - https://confluence.ihtsdotools.org/display/WIPSTS/Template+Syntax+Specification
		  ▪ 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
		
		Examples:
		
		[[+id]]: [[1..*] @my_group sameValue(morphology)] { |Finding site| = [[+id (<<123037004 |Body structure (body structure)| MINUS << $site[! SELF]) @site]] , |Associated morphology| = [[+id @my_morphology]] }
		
		• Implementation feedback on draft updates to Expression Template Language syntax
		  ○ Use cases from the Quality Improvement Project:
		  ▪ Multiple instances of the same role group, with some attributes the same and others different. Eg same morphology, potentially different finding sites.
		
		Note that QI Project is coming from a radically different use case. Instead of *filling* template slots, we're looking at existing content and asking "exactly *how* does this concept fail to comply to this template?"
		
		For discussion:
		
		[[0..1]] { [[0..1]] 246075003 |Causative agent| = [[+id (< 410607006 |Organism|) @Organism]] }
		
		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.
		
		Road Forward for SI
		
		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.
		
		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.
		
		FYI Michael Chu

Description Templates	Kai Kewley	• **ON HOLD**
		• Previous discussion (in Malaysia)
		  ▪ Overview of current use
		  ▪ Review of General rules for generating descriptions
		  • 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

Query Language - Summary from previous meetings	Linda Bird	**FUTURE WORK** **Examples: version and dialect** ○ << 64572001 |Disease| {{ term = "*heart*" }} **VERSION** http://snomed.info/sct/900000000000207008/version/20180131 ○ << 64572001 |Disease| {{ synonym = "*heart*" }} **VERSION** http://snomed.info/sct/90000000000207008/version/20180131 ○ << 64572001 |Disease| {{ FSN = "*heart*" }} **VERSION** http://snomed.info/sct/90000000000207008/version/20180131 ○ << 64572001 |Disease| {{ FSN = "*heart*" }} **VERSION** http://snomed.info/sct/90000000000207008/version/20180131, **DIALECT** W ○ << 64572001 |Disease| {{ preferredTerm = “*heart*”}} **VERSION** http://snomed.info/sct/90000000000207008/version/20180131, **DIALECT** Y ○ << 64572001 |Disease| {{ acceptableTerm = “*heart*”}} **VERSION** http://snomed.info/sct/90000000000207008/version/20180131, **DIALECT** Y ○ (* {{ term = "*heart*" }} **VERSION** http://snomed.info/sct/90000000000207008/version/20180131, **DIALECT** Z) MINUS (* {{ term = "*heart*" }} **VERSION** http://snomed.info/sct/90000000000207008/version/20170731, **DIALECT** W) ○ X MINUS Y **WHERE** X = * , Y = (* {{ term = "*heart*" }}) **VERSION** http://snomed.info/sct/90000000000207008/version/20180131, **DIALECT** W **Notes** ○ Allow nested where, version, language ○ Scope of variables is inner query

File	Modified

No files shared here yet.