

2022-11-03 - SLPG Meeting

Date & Time

10:00 to 11:00 UTC, Thursday 3rd November 2022

Goals

- Agree postcoordination profile for phase 1 of the guide

Location

[Zoom meeting link](#)

Attendees

- Chair: [Anne Randorff Højen Kai Kewley Alejandro Lopez Osornio](#)
- Attendees:
- Staff:

Agenda and Meeting Notes

Description	Owner	Notes
Welcome and agenda	All	
Postcoordination guidance	Implementation Team	Aims for this year

Postcoordination Profile Levels	Implementation Team	Agree recommended postcoordination profile for phase 1 • Level 1 - TS expects expressions to be fully MRCM compliant • Level 2 - TS permits some non-MRCM compliant expressions for limited, predefined patterns • Level 3 - TS uses generic transformations, but gives warning/errors if transformation is "risky" (note: requires MRCM compliance after transformation) • Level 4 - TS uses generic transformations, but only gives warning/error if expression is not MRCM compliant after transformation Discussion: DK: Concern that once above level 1 transformation may happen without the user wanting it - because Close-to-User-Form and Classifiable-Form look the same. Mitigation options: • Request could ask for transformation? How would this fit in with FHIR? • Syntax could include definition status (e.g. "===") explicitly when transformation not wanted? Would fit with FHIR. ◦ <i>The group like this option</i> • Something in the response to detail any transformation? • Separate endpoint. Expression communication - which form to send for interoperability? • Send classifiable form? (After transformation) ◦ Receiving system wouldn't have to support the same level of transformation • Send close to user form? ◦ More robust against modeling change ◦ Could be interpreted manually • Must send the code system version that the expression repository is dependent on Transformation documentation • How should we express the agreed transformations? ◦ Pairs of templates? ▪ Limits of capability not known ◦ How else? ▪ English and pseudocode • <i>Group preferred this option</i> • How do we categorize a transformation as level 2? ◦ Only transformations that can never go wrong (an undesired transformation without an error) ◦ If the associated precondition are not all true then there is an error and the transformation fails Transformations for Phase 1, based on Level 2: • 1. Procedure + Laterality ▪ Precondition: Procedure must have exactly one attribute with a body structure. The body structure must be lateralisable (this part of the validation could happen before or after transformation). • 2. Findings + Laterality ▪ Precondition: Same as above • 3. Finding + context attribute? (More discussion needed) • 4. Any concept + specialisation of one existing attribute ◦ Precondition: The attribute must already exist with: ▪ the same or ancestor attribute type ▪ The same or supertype of the specialisation. The attribute may or may not be grouped. ◦ Allow specialisation of attribute type? Drugs model? E.g. "Active ingredient" > "Precise active ingredient". Procedure site > direct site • More to follow...
The items below are currently on hold		
URIs for language instances		

ECL v2.2 Proposal		Find the leaves of a set of concepts - example use case (find the proximal set in the international core / or in the IPS) - example: • Example use cases ◦ Proximal ancestors in a specific module: (> concept {{ C moduleId = 1234 }}) MINUS (> (> concept {{ moduleId = 1234 }})) ▪ X = "> concept {{ C moduleId = 1234 }}" ◦ Leaf nodes: < concept MINUS (> (< concept)) ▪ X = "< concept " ◦ Removing any redundant concepts (ie subsumes another concept) from a set of concepts ▪ ^ ref set MINUS (> (^ ref set)) • X = "^ ^ ref set " Find the root concepts of a set of concepts - example use case (find the proximal set in the international core / or in the IPS) - example: ◦ Example use cases for ▪ Root nodes of an extension module: (< concept {{ module = X }}) MINUS (< (< concept {{ module = X }})) • X = "< concept " ▪ Only the 'root' concepts from a set of concepts • ^ ref set MINUS (< (^ ref set)) ◦ X = "^ ^ ref set " • X MINUS (> X) • HOMEWORK - Suggest some syntax for this. ◦ leaves(X) - eg ▪ leaves (> concept {{ C moduleId = 1234 }}) ▪ leaves (< concept) ▪ leaves (^ refset) ▪ Pros: Easy to read ▪ Cons: More consistent with the long form of ECL rather than the short form ◦ L(X) - eg ▪ L (> concept {{ C moduleId = 1234 }}) ▪ L (< concept) ▪ L (^ refset) ▪ Pros: Easy to type ▪ Cons: L could mean anything? English specific. ◦ _ (X) - eg ▪ _ (> concept {{ C moduleId = 1234 }}) ▪ _ (< concept) ▪ _ (^ refset) ◦ !_ ▪ Pros: Looks like lowest / floor ▪ Cons: No equivalent highest / top symbol ◦ !< ▪ Pros: Easy to type. Familiar looking syntax. ▪ Cons: May be too similar to children of - confusing? ◦ !!< (bottom) !!> (top) ▪ Pros: Easy to type. Familiar syntax. Different enough from <! and >! ▪ Cons: None that I can think of ◦ X (bottom), X (top) ▪ Pros: Matches existing mathematical syntax for the Floor and Ceiling functions, which have similar meaning. ▪ Cons: Could be challenging for some people to type on the keyboard ◦ <!! (bottom), >!! (top) ▪ Pros: Easy to type. Familiar looking syntax. Won't be mistaken for children of. Both top and bottom can be represented clearly. ▪ Cons: Too long? (Makes operators three characters rather than two).
ECL v2.1 - Requirement proposals (to be archived)	All	Potential requirements for ECL v2.1 - Discussion and brainstorming

- **Daniel's comments**

- An explanation of the results of this ECL: <<385540001 |Olmesartan containing drug| MINUS (<<763158003 |Medicinal product (product)|: { <<127489000 |Active ingredient| = <<412259001 |Olmesartan| })
- Cardinality of focus concepts is not supported (Example use case: which concepts have 2 or more stated IsAs)
- (Related) Using 116680003 |Is a (attribute)| in refinement/dotted expressions, e. g. 395026007.116680003, 395026007:[2..*]116680003=*
Snowstorm and Ontoserver give different results for dotted expressions.
[https://snowstorm-alpha.ihtsdotools.org/snowstorm/snomed-ct/fhir/ValueSet/\\$expand?_format=json&count=10&url=http:%2F%2Fsnomed.info%2Ffhir_vs%3Decl%2F395026007.116680003](https://snowstorm-alpha.ihtsdotools.org/snowstorm/snomed-ct/fhir/ValueSet/$expand?_format=json&count=10&url=http:%2F%2Fsnomed.info%2Ffhir_vs%3Decl%2F395026007.116680003)
[https://r4.ontoserver.csiro.au/fhir/ValueSet/\\$expand?_format=json&count=10&url=http:%2F%2Fsnomed.info%2Ffhir_vs%3Decl%2F395026007.116680003](https://r4.ontoserver.csiro.au/fhir/ValueSet/$expand?_format=json&count=10&url=http:%2F%2Fsnomed.info%2Ffhir_vs%3Decl%2F395026007.116680003)
- We've done history for concepts but not for any other components...

Realizing the information on inactive concepts is different and more extensive from other components, {{ + HISTORY... }} is (just) sugar and maybe just maybe there might be reasons to sprinkle other components with this sweetness (or not).

Examples: previously active refset members (and their fields), previously active descriptions (e.g. Refers to association refset), ...

- **Context supplements** - e.g.
 - `<< 56265001 |Heart disease| {{ + CONTEXT }}` – This syntax is too general, as there is a risk of including absent finding, not-done procedure and family history
 - `<< 56265001 |Heart disease| {{ + CONTEXT-DEFAULT }}` ? – What would this mean?
 - - **Brief form:**
 - `[[@ecl_query]] {{ + Context (Temporal = [[ @temporal_value]] )}}`
 - **Expanded form:**
 - `[[@ecl_query]] OR (< 243796009 |Situation with explicit context|: { (246090004 |Associated finding| = ([[@ecl_query]]) OR |Associated procedure| = ([[@ecl_query]]) (|Procedure context| = |Done| OR |Finding context| = |Known present|), |Subject relationship context| = |Subject of record|, |Temporal context| = [[@temporal_value]] })`
 - **Example 1: `<< |Heart procedure| {{ + Context (Temporal = *) }}`**
 - `<< |Heart procedure| OR (< 243796009 |Situation with explicit context|: { 246090004 |Associated finding| = << 56265001 |Heart disease|, |Procedure context| = |Done|, |Subject relationship context| = |Subject of record|, |Temporal context| = * } )`
 - **Example 2: `(<< |Heart disease| OR << |Heart procedure| ) {{ + Context (Temporal = *) }}`**
 - `<< |Heart procedure| OR (< 243796009 |Situation with explicit context|: { (246090004 |Associated finding| = (<< |Heart disease| OR << |Heart procedure|) OR |Associated procedure| = (<< |Heart disease| OR << |Heart procedure|)) (|Procedure context| = |Done| OR |Finding context| = |Known present|), |Subject relationship context| = |Subject of record|, |Temporal context| = * })`
 - `<< 56265001 |Heart disease| {{ + Context (Temporal = *, FindingContext=<<|Known present| )}}`
 - Will return all types of heart disease, plus concepts like 394886001 |Suspected heart disease (situation)|, and 429007001 |History of cardiac arrest (situation)|
 - Expands to:
 - `<< 56265001 |Heart disease| OR (< 243796009 |Situation with explicit context|: { 246090004 |Associated finding| = << 56265001 |Heart disease| } )`
 - However, you may want to exclude (or include) specific contexts - for example:
 1. To ensure that the finding was about the subject of the record (and not a family history, e.g. to exclude 429959009 |Family history of heart failure (situation)|), you could say:
 - `<< 56265001 |Heart disease| {{ + CONTEXT (relationship = 410604004 |Subject of record| )}}`
 2. To ensure that the finding was 'Known present' (e.g. to exclude 394926003 |Heart disease excluded (situation)|), you could say:
 - `<< 56265001 |Heart disease| {{ + CONTEXT (finding_context = << 410515003 |Known present| )}}`
 3. To ensure that the finding was about the subject of the record AND known present, you could say:
 - `<< 56265001 |Heart disease| {{ + CONTEXT (relationship = 410604004 |Subject of record|, finding_context = << 410515003 |Known present| )}}`
 4. ?? Is there any use case for restricting adding temporal context? (e.g. `temporal != << 410513005 |In the past|`)
 - Is any more syntactic sugar required? E.g.
 - `{{ + CONTEXT (relationship = self, finding context = present, temporal != past) }}`
 - `{{ + CONTEXT (self, present, ! past) }}`
 - Other ideas? Common profiles?
- -----
- **Ability to return attribute types** (see proposal below)
 - `[ attributes ] << 125605004 |Fracture of bone (disorder)|`
 - `<< 125605004 |Fracture of bone (disorder)| . Attributes`
 - `<< 125605004 |Fracture of bone (disorder)| . (<< 125605004 |Fracture of bone (disorder)| . Attributes )`
 - `[ attribute, value ] << 125605004 |Fracture of bone (disorder)|`
- -----
- **Reverse membership** (see below)
 - Which reference sets "contain" the given concept(s) - e.g. 421235005 |Structure of femur|?
 - 421235005 |Structure of femur| . Refsets
 - 421235005 |Structure of femur|. Refsets [referencedComponentId]
 - 421235005 |Structure of femur| . Refsets [targetComponentId]
- -----
- Other?

Returning Attributes	Michael Lawley	- 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: <pre><< 404684003 Clinical finding : 363698007 Finding site = <<66019005 Limb structure </pre> <pre><< 404684003 Clinical finding . 363698007 Finding site </pre> 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	What resets 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
Postcoordination Topics		- Discuss feedback on transformation implementation - Resources Expression transformation service generates the classifiable form and the necessary normal form from a close to user form expression - Contribute your expression examples and write your feedback for consideration and discussion - Recap of SNOMED on FHIR discussions - What is the functionality scope of a terminology server that supports postcoordination? For example, does it include: - Classifying multiple expressions in a single substrate? What are the use cases for this? - Assigning (local) identifiers to expressions? What are the use cases for this? - Autogenerating or assigning a term to an expression? What are the use cases for this? - Does a terminology server that supports postcoordination, include all the functions of an expression repository? - What is the relationship between a terminology server that supports postcoordination, and an expression repository? - Outstanding questions - What are the pros and cons of extending SCG to allow an expression as the focus of a postcoordinated expression? - Note: This was raised in context of a NNF generated over a postcoordinated substrate, where the proximal parent is an expression - Example of using expressions in focus concept (125605004 Fracture of bone :363698007 finding site = 84167007 Foot bone) : 272741003 Laterality = 7771000 Left 125605004 Fracture of bone :363698007 finding site = 84167007 Foot bone , 272741003 Laterality = 7771000 Left - What is the expected NNF when classifying an expression that is equivalent to a precoordinated concept? For example: Expression that is equivalent to 111273006 Acute respiratory disease 64572001 Disease (disorder) : {263502005 Clinical course (attribute) = 424124008 Sudden onset AND/OR short duration (qualifier value) } {363698007 Finding site (attribute) = 89187006 Airway structure (body structure) } - Options: 111273006 Acute respiratory disease : {263502005 Clinical course = 424124008 Sudden onset AND/OR short duration } {363698007 Finding site = 89187006 Airway structure } 50043002 Disorder of respiratory system (disorder) + 2704003 Acute disease (disorder) : {263502005 Clinical course = 424124008 Sudden onset AND/OR short duration } {363698007 Finding site = 89187006 Airway structure } - Other? - Recap of internal discussions with Content Team - Inter-attribute dependencies - Grouping rules

Dynamic Templates		- Continue discussion on dynamic templates - Inter-attribute dependencies Acute/Chronic and Inflammation - Adding a clinical course requires specializing the inflammation morphology ? - E.g. Pyelonephritis : Clinical course = Chronic should be Pyelonephritis : Clinical course = Chronic , Associated morphology = Chronic inflammation - E.g. Pyelonephritis : Clinical course = Sudden onset AND/OR short duration should be Pyelonephritis : { Clinical course = Sudden onset AND/OR short duration }, Associated morphology = Acute inflammation Infectious Causative Agents - Adding a causative agent = Domain Bacteria or Virus requires adding a Pathological process = Infectious process - E.g. Nephritis : Causative agent = Domain bacteria should be Nephritis : Causative agent = Domain bacteria , Pathological process = Infectious process Congenital and Acquired - Adding an Occurrence of Congenital to a focus concept with an abnormal morphology, requires adding a Pathological process of Pathological development process - E.g. Koilonychia : Occurrence = Congenital should be Koilonychia : Occurrence = Congenital , Pathological process = Pathological developmental process Situations with Explicit Context - if the procedure context = Planned , then the temporal context should be << Current of specified time - If the procedure context = In progress , then the temporal context should be << Current - If the procedure context = Performed or Done , then the temporal context should be << Current or past (actual) - Note: for this use case (of Procedure with explicit context) perhaps we just recommend (or require) that the full role group is spelled out. - Next steps - Representation of the content rules - Who creates the complete list of rules and how? - What formalism? - Determine which are mandatory and which are optional - Implementation of content rules - e.g. - Guided data entry by pre-populating role groups in expression template based on definition of focus concepts (for design-time use, such as mapping) - Mandatory content rules could be added to transform process
Postcoordination Use Case Examples	All	Example 1 - Dentistry / Odontogram - Requires an expression template to create expressions. - Resulting expression still requires a transformation to make it classifiable Example 2 - Terminology binding - Uses a fixed expression template to combine codes entered into separate fields - The procedure+laterality example still requires a transformation to make it classifiable Example 3 - Mapping - Design-time activity - Map targets may not be able to be fully represented using concept model attributes - In many cases, an extension (with primitive concepts) should be recommended where there are gaps in the mapping - There may be some cases in which postcoordination is helpful (e.g. LOINC to SNOMED CT map) Example 4 - Natural Language Processing - Usually run-time activity. - May require manual confirmation of coding suggestions (unless low clinical risk, eg for suggesting relevant patient records for manual review)
Postcoordination Guidance	Anne Randorff Højen , Kai Kewley	Practical Guide to Postcoordination

- **Proposal - Use syntax (i.e. braces) to distinguish refinement vs new role group**
- There should be a syntactic distinction between refinement and constructive addition (ie adding a new role group). That is:
 1. 83152002 |Oophorectomy| : 405815000 |Procedure device| = 122456005 |Laser device|
 - is classified as (i.e. the refinement is added to the role groups in the definition of the focus concept(s)):
 - 83152002 |Oophorectomy| :
 - { 260686004 |Method| = 129304002 |Excision - action|,
 - 405813007 |Procedure site - Direct| = 15497006 |Ovarian structure|,
 - 405815000 |Procedure device| = 122456005 |Laser device| }
 - 2. 83152002 |Oophorectomy| : { 405815000 |Procedure device| = 122456005 |Laser device| }
 - is classified as:
 - 83152002 |Oophorectomy| :
 - { 260686004 |Method| = 129304002 |Excision - action|,
 - 405813007 |Procedure site - Direct| = 15497006 |Ovarian structure|,
 - { 405815000 |Procedure device| = 122456005 |Laser device| }
 - However, for attributes which are **always** self-grouped - i.e. Priority, Due to, After, Before, During, Clinical course, Temporally related to, and all Observable entity attributes (see [Relationship Group](#)), these must always be put into their own role group:
 - 1. 125605004 | Fracture of bone | : 42752001 | Due to (attribute) | = 1912002 | Fall |
 - is classified as:
 - 125605004 | Fracture of bone | : { 42752001 | Due to (attribute) | = 1912002 | Fall | }
 - or
 - 125605004 | Fracture of bone | :
 - { 363698007 |Finding site| = 272673000 |Bone structure|,
 - 116676008 |Associated morphology | = 72704001 |Fracture| }
 - { 42752001 | Due to (attribute) | = 1912002 | Fall | }
- **Proposal: Expression forms needed for this** (see [3.4 Transforming Expressions](#))
 - Close to user form - e.g. 83152002 |Oophorectomy| : 405815000 |Procedure device| = 122456005 |Laser device|
 - Canonical close to user form - e.g. 83152002:405815000=122456005
 - Classifiable form (SCG) - e.g. 83152002:{260686004=129304002,405813007=15497006,405815000=122456005}
 - PLUS Classifiable form (OWL) - e.g.
 - EquivalentClasses(:123063
 - ObjectIntersectionOf (:71388002
 - ObjectSomeValuesFrom(:609096000 ObjectIntersectionOf(ObjectSomeValuesFrom(:
 - 260686004 :129304002)
 - ObjectSomeValuesFrom(:405813007 :15497006))))
 - Necessary normal form - e.g. 83152002+416376001:{260686004=129304002,405813007=15497006,405815000=122456005}
 - PLUS Necessary normal form (tables)
 - Relationships:
 - (123063 116680003 83152002) - 0
 - (123063 260686004 129304002) - 0
 - (123063 405813007 15497006) - 1
 - (123063 405815000 122456005) - 1
 - Primitive expressions - "<<<" (only useful in a mapping context) relies on the assigned identifier (which are necessarily semantically unique).
- **Proposed Transformation Rules - Refinements (in valid domain of focus concepts)**

Close-to-user-form - IF the grouping of the refinement is **not** concept model valid THEN

If there is a single (non-self-grouped) role group in the definition of the focus concept, then any ungrouped (but groupable) refinements are merged with this role group

If there is more than one (non-self-grouped) role group in the definition then flag as ambiguous and require refinement

[NEED TO FIND a realistic clinical example where this may occur // Prevent failing cases from coming up // use template](#)

[ALTERNATIVE: Refinement is applied to all \(non-self-grouped\) role groups in the definition](#)

Self-grouped attributes in the refinement are grouped on their own - i.e. Priority, Due to, After, Before, During, Clinical course, Temporally related to, and all Observable entity attributes (see [Relationship Group](#))

Self-grouped attributes in the definition of the focus concept(s) are left unchanged

- 1. **Single refinement**
 - 83152002 |Oophorectomy| : 405815000 |Procedure device| = 122456005 |Laser device|
 - 83152002 |Oophorectomy| :
 - { 260686004 |Method| = 129304002 |Excision - action| ,
 - 405813007 |Procedure site - direct| = 15497006 |Ovarian structure| ,
 - 405815000 |Procedure device| = 122456005 |Laser device| }
- 2. **Two groupable refinements**
 - 83152002 |Oophorectomy| : 405815000 |Procedure device| = 122456005 |Laser device| , 363700003 |Direct morphology| = 367643001 |Cyst |
 - 83152002 |Oophorectomy| :
 - { 260686004 |Method| = 129304002 |Excision - action| ,
 - 405813007 |Procedure site - direct| = 15497006 |Ovarian structure| ,
 - 405815000 |Procedure device| = 122456005 |Laser device| ,
 - 363700003 |Direct morphology| = 367643001 |Cyst | }
- 3. **One groupable refinement with one self-grouped refinement**
 - 83152002 |Oophorectomy| : 405815000 |Procedure device| = 122456005 |Laser device| , 260870009 |Priority| = 394849002 |High priority|
 - 83152002 |Oophorectomy| :
 - { 260686004 |Method| = 129304002 |Excision - action| ,
 - 405813007 |Procedure site - direct| = 15497006 |Ovarian structure| ,
 - 405815000 |Procedure device| = 122456005 |Laser device| }
 - { 260870009 |Priority (attribute)| = 394849002 |High priority| }
- 4. **Refinement attribute matches (or subsumed by) attribute in focus concept's definition**
 - 83152002 |Oophorectomy| : 260686004 |Method| = 277261002 |Excision biopsy (qualifier value)|
 - 83152002 |Oophorectomy| :
 - { 260686004 |Method| = 129304002 |Excision - action| ,
 - 260686004 |Method| = 277261002 |Excision biopsy (qualifier value)| ,
 - 405813007 |Procedure site - direct| = 15497006 |Ovarian structure| }
- 5. **Refinement explicitly in role group**
 - 83152002 |Oophorectomy| : { 260686004 |Method| = 281615006 |Exploration - action| , 405813007 |Procedure site - direct| = 367643001 |Cyst | }
 - 83152002 |Oophorectomy| :
 - { 260686004 |Method| = 129304002 |Excision - action| ,
 - 405813007 |Procedure site - direct| = 15497006 |Ovarian structure| ,
 - { 260686004 |Method| = 281615006 |Exploration - action| ,
 - 405813007 |Procedure site - direct| = 367643001 |Cyst | }
- **Proposed Transformation Rules - Refinements (NOT in valid domain of focus concepts)**

Close-to-user-form - IF the refinement's attribute is **not** valid for the domain of the focus concept THEN

If there is a single role group in the definition of the focus concept, which has an attribute value in the domain of the refinement's attribute THEN nest the relevant attribute value with the refinement added to the attribute value

(Note: It doesn't matter if the role group is self-grouped or not (see example 1 below)

If there is more than one role group in the definition of the focus concept, which has an attribute value in the domain of the refinement's attribute THEN (non-self-grouped) role group in the definition then flag as ambiguous and require refinement

- 1. **Left aural temperature**
 - 415974002 |Aural temperature| : 272741003 |Laterality| = 7771000 |Left|
 - 415974002 |Aural temperature| : {704327008 |Direct site| = (42859004 |Ear drum| : 272741003 |Laterality| = 7771000 |Left|)}
- 2. **Malignant tumor of right ovary**
 - 363443007 |Malignant tumor of ovary| : 272741003 |Laterality| = 24028007 |Right|
 - 363443007 |Malignant tumor of ovary| :
 - { 116676008 |Associated morphology| = 367651003 |Malignant neoplasm of primary, secondary or uncertain origin| ,
 - 363698007 |Finding site| = (15497008 |Ovarian structure| : 272741003 |Laterality| = 24028007 |Right|) }
- **Other Example - Emergency excision of appendix**
 - 1. 80146002 |Excision of appendix| :
 - 260870009 |Priority| = 25876001 |Emergency|
- **Other Example - Fracture of bone**
 - 1. 125605004 |Fracture of bone| : 363698007 |finding site| = 84167007 |Foot bone|
 - 2. 125605004 |Fracture of bone| : {363698007 |finding site| = 84167007 |Foot bone| }
 - 3. 125605004 |Fracture of bone| : {116676008 |Associated morphology| = 72704001 |Fracture| , 363698007 |finding site| = 84167007 |Foot bone| }
 - 4. 64572001 |Disease| : {116676008 |Associated morphology| = 72704001 |Fracture| , 363698007 |finding site| = 84167007 |Foot bone| }

The items below are currently on hold

URIs for Extended Editions		ON HOLD - 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)
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 <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?" 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
--------------------------	----------------------------	---

File	Modified
Microsoft Powerpoint Presentation SNOMED Languages Project Group (September 2022 business meetings).pptx	2022-Oct-31 by Anne Randorff Højen

Agenda and Meeting Notes