Mechanisms for working with Languages

HL7 Vocab Item: https://gforge.hl7.org/gf/project/fhir/tracker/?action=TrackerItemEdit&tracker_item_id=15806

ML: Supplement would hopefully only add additional descriptions that are not described in the base content.

Precedence (fallback) for multiple language reference sets only really discussed in Languages Group for ECL - other use cases not yet brought to light.

Update 28 Aug 2018: Keen to see an implementation using Supplements to see how it would actually work. Option to explore this at Baltimore Connectathon?

Update 11 September 2018: Do we need a parameter for the expand operation for this - Reuben looking into this currently.

Update 11 Dec 2018: HTTP Headers RFC 5646 (Tags for identifying languages). Suggestion that where dialects are to be referenced which do no have a standard code, one could be constructed, either using new letters or the language reference set id directly eg en-nz-x-12345601 Note also that requested language/dialect could be an ordered list for gradual fallback where descriptions may or may not be available.

Can also be passed in the language parameter passed to an expansion operation.

Language reference set could be an implicit code system supplement.

Update 8 Jan 2019: Note that SNOMED CT Language Reference sets do not indicate if they relate to a dialect, language or context of use. GG suggested FHIR could address that more directly than using a reference set SCTID to supply that desired context. Include someone in a non-English speaking country!

Update 5 March 2019: Reuben suggesting plan of action to ensure we have a solid proposal to take to the Montreal Connectathon (4-5 May 2019). Languages working group code for "consumer friendly terms" - Rob says that a harmonization proposal is in the pipeline

Michael Lawley has raised ticket about the "use" field being limited to FSN/Synonym. Elsewhere in FHIR there is a "display" code that can be used to indicate other languages See 22490. Also 19960 - additional term for "Consumer Terms" ready for implementation RS (Q4 2020 at the earliest).

18 June PWI gave some notes from pop-up session at FHIR Dev Days - more about locale than language. ML: "Translation" extension doesn't allow for a particular piece of text being in a different language from that of the resource.

Peter Jordan's implementation can take a refset Id (which is more flexible for dealing with same-language designations such as dialect and patient friendly terms "includeDesignation = SYSTEM+<SCTID>")

Problem: We can't tell the difference between Preferred and Acceptable. Currently the best we can hope for is to use array ordering.

Update 8 October 2019: Do we need a way to pass a refset id to designation use in an expansion operation? The expansion operation says it will accept "A token that specifies a system+code that is either a use or a language" Suggestion that we could then return the SCTID for the particular language refset in ValueSet.compose.include.concept.designation.use This extended use of that field doesn't give us the ability to say Acceptable or Preferred, but perhaps it's sufficient. We could potentially use ordering of same-refset items to put the preferred terms first. Note that this is very SNOMED specific and doesn't offer a solution for patient friendly terms eg for LOINC.

DK "Acceptability" sounds like an extension that this group could propose.

Update Nov 12 Recommended option for specifying a language refset:


This is based on the definition of the designation parameter: "A token that specifies a system+code that is either a use or a language. Designations that match by language or use are included in the expansion. If no designation is specified, it is at the server discretion which designations to return" A token is specified as <system> | <code>
In terms of the designations returned from, say, lookup, the group felt that it was reasonable to suggest that any given designation could have more than one use, and we would like to say something about its language reference set entry as part of some additional "use" element.

7 Jan 2020: In this case the "use" could be used to specify the language reference set SCTID that was relevant, and so we might in fact see the same description appear more than once if referred to by multiple language reference sets. An extension which allows for specifying the additional two pieces of information (language refset and acceptability) might be preferable to modifying the existing standard regarding "use". This would include a corresponding filter parameter. Also check that multi-language editions behave sensibly when specifying both a language code and a language reference set.

Example of current implementation:

```json
{name: "designation",
 part: [{name: "language", valueCode: "en"},
    {name: "use", valueCoding: {
      system: "http://snomed.info/sct",
      code: "900000000000013009",
      display: "Synonym"}},
    {name: "value", valueString: "Obstetric umbilical artery Doppler" }]
}
```

Question outstanding: if an acceptable designation is not available in the specified language reference set, would we want to return the fall back (default) language refset instead? In that case we would absolutely need to be able to indicate in which language reference set the term is acceptable/preferred.

Also, can we specify in the request that we are only interested in preferred terms, rather than current implementation - both acceptable and preferred.

Also, the filter currently works at the level of the concept and the specified language. Could we filter in a way that was specific to the designation use?

Thinking aloud, shoe-horning everything into one block and using a draft URI proposal:

```json
valueCoding: {
  system: "http://snomed.info/sct", --No need to specify http://snomed.info/syntax/scg because a PCE is still a SNOMED code.
  code: "900000000000013009+900000000000508004+900000000000548007",
  display: "Synonym+GB Language+Preferred"}
```

Update 26 November: The semantics of this post coordinated expression are problematic. Is a SNOMED specific extension the way to go here? Firstly by working out the most useful value for "use" and then add further detail into the extension. Is 'Patient Friendly' part of the R5 specification yet?

Use case - querying for terms in a particular language reference set. When not all of the terms are part of that language reference set (eg no patient friendly term exists) we want to indicate which ones are patient friendly and which ones are not. OR should we return nothing when no acceptable/preferred description is available and allow the client to make a follow-up call requesting the "fallback position" language reference set - like US English. This behaviour would reduce the need for an extension. Would we return the concept without a designation in that case - so the client could detect that there is more information to be obtained.

We still have no way to differentiate between preferred terms and merely "acceptable".

Some indicator of "Not Acceptable" would allow us to indicate non-membership of the specified language reference-set.

$lookup is more problematic as a number of designations are returned and no indication is given for their context of use. Perhaps a stronger use case for the need for an extension.

Update 10 December: Peter G. Williams to try to have language refset token implemented for Sydney Connectathon.

In terms of the 3 values we're trying to represent, do we need to suggest an extension to FHIR? ML: What are the client use cases - could the SNOMED UI Demo be enhanced to support variable langRefsets? Knowing which terms are patient friendly is possibly not needed in this case.
If language refset is requested and no term is acceptable/preferred is returning the SCTID *without* the term an option? DK In the case of patient friendly terms, any term would be preferable to nothing.


RH: [https://jira.hl7.org/browse/FHIR-19960](https://jira.hl7.org/browse/FHIR-19960)

Note that "Patient Friendly" can also exist in many languages - these are separate dimensions.

**TODO** Look at 160 PF terms in NZ Edition for January.

**Update 11 Feb**

At the Sydney Connectathon we received the advice that the token solution described above was sub-optimal and that we should access the features required using an extension. See subpage: