

URI Proposal: Addition of xsct

Background

The FHIR API necessarily assumes that any given SNOMED CT concept has been published and appears in a specific CodeSystem instance, ie that it *actually officially exists*. In practice, however, a SNOMED Terminology Server used for the creation of SNOMED CT concepts will feature some sort of workflow of "work in progress", unpublished concepts. In this situation there will be a requirement to access this unpublished content.

As an example, in the specific situation of the Snowstorm implementation, terminology author's work is organised into branches much like a developer's code repository with official releases being held on a branch off the "root" like MAIN/20210131, and the "daily build" existing on the root itself. By default any FHIR API request will run against a particular release branch (or the most recent published version if not otherwise specified) and so we need a way to indicate that the request should run against the work-in-progress, root, daily-build branch.

This FHIR based requirement and solution has a very narrow focus. Unfortunately, the implications of such a proposal are much broader.

Reusing "xsct"

SNOMED International already uses the X (eXperimental) indicator for alpha & beta releases of International and Country Editions of SNOMED CT
eg xSnomedCT_BelgiumExtensionRF2_PREPRODUCTION_20210315T120000Z/Snapshot/Terminology/xsct2_Concept_Snapshot_BE1000172_20210315.txt this additional letter is added to make clear that the contents have not been "officially" published, and components are often marked with a (future) effective time.

Other Use Cases

There is a use case for making content available externally before it can be officially published with COVID-19 Vaccines being a real-world example of this.

Context(s) of use

There are a number of places in which an 'x' modified URI could be used:

1. Referring to the unpublished content (ie daily build, work-in-progress)
2. Referring to a complete 'Member release' or 'Alpha/beta release' loaded into a terminology server.

What's quite fortunate between these two use cases is that - as understood within SNOMED International - an alpha or beta package looks just like an officially published release. The effective time is populated where required, and will be the intended effective time of the eventual finished product. In the SI release process, this may be produced from the daily build / work-in-progress .

In the context of FHIR, most operations specify a URI for the code system itself. This would usually be <http://snomed.info/sct> and so an "unversioned" variant of that would be <http://snomed.info/xsct> However we should also consider if the 'x' could be applied to component URIs eg <http://snomed.info/xid/288804006> (see [2.2 URIs for Components and Reference Set Members](#)) and with the module and version also specified <http://snomed.info/xsct/900000000000207008/version/20210131/id/288804006>

Note that - as far as FHIR is concerned for example - a SNOMED concept is a SNOMED concept. It would be quite acceptable to refer to <http://snomed.info/id/65781000052101> a concept in the Swedish edition without the module - <http://snomed.info/sct/45991000052106/id/65781000052101>

Implications

Given that Alpha/Beta/Preview releases are populated with an effective time, it *should* normally be quite possible to refer to them using a normal URI, even if the effective time is in the future.

If possible, any response returned from a query that includes this 'unpublished' indicator should include a warning that the SCTIDs are subject to change until the point where they have been officially published. They are for testing and evaluation purposes only, and should on no account be used in any Patient Medical Record.

Alternative Solutions

Previous suggestions for accessing unpublished content included using the string "UNVERSIONED" in place of an effective time in the version URI:

<http://snomed.info/sct/900000000000207008/version/UNVERSIONED>

This visually clunky solution is at least unambiguous and not easy to miss.

