

Features

Overview

Documents features of the system based on the proposal submitted for design/development.

Non-functional Features

- 100% web-based and accessible through any modern web browser (does not include older versions of IE).
- The project structure allows for builds of the tool that allow developers to customize various aspects of the user interface for rebranding purposes. These include (but are not limited to)
 - Application header, including icon, title, and links.
 - Application footer, including copyright information, links, and version information.
 - Authentication screen (including username/password if needed with customizable “forgot my password” links).
 - Intro text (when viewing the page prior to logging in).
- The server architecture uses a variety of configurable handler-based mechanisms that support customizable components for things like authentication/authorization, identifier management, preferred name computation, and import and export formats.
- The server architecture supports pluggable implementations of core interfaces, allowing for persistence to be implemented either locally, or via a connected service, such as the IHTSDO Terminology Server. Thus the application can use entirely local storage if the user does not have access to an appropriate server environment, or the application can use entirely remote storage if the supporting APIs are available in the underlying terminology server product.
 - Some things like the ability to simultaneously manage multiple versions of a reference set and support queries into the past (to see what reference set versions used to look like) may not be inherently supported in underlying terminology server environments.
- This solution is specifically designed to be able to work for IHTSDO itself and be reconfigurable and rebrandable for any member or affiliate organizations that desire to use it.
- The standard operating environment for this tool includes only freely available, open-source solutions

Refset Features

- Create extensional reference sets through any of these mechanisms:
 - Loading an initial set of members from an RF2 (simple reference set) file.
 - Computing an initial set of members from a query (or combination of queries)
 - Manual entry based on known codes, text-based searches, or hierarchical searches (e.g. via the IHTSDO browser).
 - Loading directly from the IHTSDO terminology server (to the extent that this capability is supported). This is important because it allows previously developed referenced sets to automatically be pulled into the tool for maintenance.
- Maintain extensional reference sets. This mostly includes the ability to identify individual members of the reference set that have become obsolete with an update to the underlying SNOMED CT® edition used to create it. It also includes the ability to manually remove entries or to add them using any of the mechanisms described in the previous bullet.
 - The tool will provide an on-demand feature for migrating a reference set to the new version. The author will choose the new edition of SNOMED CT® to update to – which may be a newer version, or may be an entirely different edition (to handle the case of say moving a maintained reference set from an NRC to the core, or vice versa).
 - The validation framework (described below) will be employed at this time to perform validation checks on all members against the updated SNOMED CT® version. Things like members now obsolete will be flagged and the author given an opportunity to make relevant changes.
 - At any time during this process
- Create intensional reference sets by defining a set of queries (in the grammar of the expression constraint language supported by the terminology server) that produce results from the IHTSDO terminology server. Also support manual override with “include” and “exclude” lists for outlier codes.
 - Ideally, in the long run, the grammar used to support this tool would converge with the emerging “Expression Constraint Grammar” being developed by IHTSDO.
- Maintain intensional reference sets. This includes both the ability to materialize the definition against the new version of SNOMED CT® as well as to identify manually edited members that are no longer valid.
 - As with extensional refset maintenance, this will be an on-demand feature that will include running validation checks against all individual members of the refset.
 - It will also include the ability to visualize changes in members based on evaluating the definition against a different edition of SNOMED CT®. Authors will be able see potential new members, potential deprecated members, and be able to accept or reject these changes.
- Clone an intensional or extensional refset. An important use case is starting with an existing reference maintained by another organization or SNOMED CT® edition and tailoring it to work for yours. For example, the US may want to start with a UK reference set and then customize it for use with the US edition of SNOMED CT®.
- Convert an extensional reference set to an intensional one by extrapolating a minimum-spanning query that covers the content involved (based on “isa” hierarchies with boolean AND expressions to include outliers and NOT expressions to exclude outliers).
 - Internally, this means that all reference sets can be inherently maintained as “intensional”, which likely will lead to better maintenance of otherwise extensional reference sets and also highlight areas of SNOMED CT® content that may need improvement.
- Import and export of extensional reference set definitions (in the form of an RF2 reference set itself).
- Remove an intensional or extensional reference set when no longer desired or needed.
- Tracking of reference set metadata, including its version, type, the edition of SNOMED CT® that was used to create it, an indicator of whether it has been published, an indicator of its workflow status if still in development, and the authors who are currently associated with its development and maintenance (and their corresponding roles).
 - Also includes the ability to define an “external” reference set that is simply a pointer to an external URI and does not have any materialization in this tool.

- Generalized facility for comparing two reference sets. This would involve a user interface that presented a “diff” style report in which members in common, members added, and members removed were all clearly shown with the ability to refine the reference set metadata on either side of the comparison to iterate through comparisons to converge on a solution. This feature is intended to support these use cases (among others):
 - Comparison of a published reference set against a particular version of SNOMED CT® against a future version (e.g. the next version).
 - Comparison of a published reference set of a particular edition of SNOMED CT® against a different edition (e.g. the international version of a reference set vs. the US version).
 - Comparison of an unpublished reference set against a prior version of that reference set (e.g. to understand the implication of changes in an intensional reference set definition on the overall result).
 - NOTE: this relies on the ability of the underlying terminology server to support the ability to view different forms of the reference set at different times in its history. If this is not supported, then these features will have to be managed and persisted locally by the application.
- Simple workflow mechanism to track reference sets that are being developed, are ready for review, are reviewed, are ready for preview by a wider (and possibly public) community, are ready for publication, and finally are actually published.
 - Overall workflow state of a reference set is tracked in its metadata. Behaviour regarding the visibility of this reference set to users with different roles can be controlled (e.g. to only show publicly a reference set that is intended for preview).
 - Role-based access to changing of the workflow state of a reference set.
 - Support for a simple reviewer (non dual independent review) workflow.
 - Track history of official releases of the refset (so they can be easily referenced)
- Configurable email address for feedback on a reference set during preview and publication modes.
 - Sophisticated within-application feedback features are not included at this time.
 - This mechanism allows individual organizations to set the feedback email for their organization to support a local ticketing system (e.g. Freshdesk, Siebel, or just a basic email account checked on a regular basis).
- Facility for search and retrieval of the entire catalog of available reference sets. Role-based access is included so that reference sets can be made private, or only accessible to users with particular roles for a particular reference set (or project for multiple reference sets).
 - All reference set metadata is searchable.
 - Individual computed reference set members are searchable.
 - Retrieval can support visualizations of past versions of the reference set (and/or comparisons to current ones).
 - Search results separate reference sets that a particular user has a non-viewer role on, reference sets available for preview, and reference sets that have been published.
- Support for attaching related artifacts to a “published” reference set to allow the tool to be used as a distribution portal (if desired). This would include:
 - Documentation
 - Release files representing publication state (in RF2).
 - Release files representing intensional reference set definitions (in RF2).
 - Ancillary or auxiliary attachments.
 - Human readable form of the reference set will be a URL that directs a user into the tool where the reference set itself can be visualized, browsed, and searched.
- Clear division between the “information portal” and “development” aspects of this tool. Users who are interested in discovering, searching, or browsing publicly available and published or preview reference sets can interact exactly in that way. They can learn about, download, visualize, and provide feedback on these various things. Authors who are actively developing and maintaining reference sets will have access to different features of the tool that allow for testing of variations in intensional reference set definitions, manual including or excluding of specific codes, comparison of reference sets in a variety of ways, and access to the underlying reference set workflow.
- Project-based mechanism to allow for grouping the maintenance of multiple reference sets under the same “organization”. This streamlines user-role management and binds together various reference sets that may belong to or be maintained by the same organization.
- Validation framework for identifying and preventing undesirable data conditions (such as publication of a reference set that has members that are inactive in the SNOMED CT® edition backing the reference set).
 - Supports errors to prevent moving forward in workflow.
 - Supports warnings that allow editors to “click through” if they know the data condition to be correct.

Translation Features

- Translation will be implemented as a module that extends the functionality of what can be done with a reference set given the previous sections. This means that defining the pool of content to be translated is done by developing or importing a reference set. Each “version” of a translation will be correlated to a “version” of the reference set.
- After defining or loading and then creating a “translation reference set”, the tool will provide the ability to initiate a translation project. Existing content can be loaded into that translation project from RF2 files (e.g. descriptions and language reference sets). While the import handler is extensible, the default implementation will only support RF2. We are of the opinion that conversion of other formats to RF2 is relatively trivial and keeps operation of the system less complicated.
- Tracking of translation project metadata, including the reference set used to define it, the spelling correction dictionary it will use, and information like allowable description types (and their definitions), and information about releases. User and role pairings are associated with the project used to create the reference set that backs the translation project – thus no special need exists for user /role management at this level.
- Export of completed translations to RF2 (descriptions and language reference set files). The tool will use an extensible handler mechanism with a default RF2 implementation that could be customized by members if another format was needed.
- Simple workflow mechanism that supports these features:
 - Assignment of batches to authors. All content to translate will exist as a single collection of concepts that can be sorted, searched, and filtered to produce the set of concepts desired to be assigned to a particular author.
 - Ability to develop translations for one or more descriptions of a concept. When a concept is presented to an author for translation, all descriptions of the concept to translate will be shown with an option for that author to translate one or more. Validation checks can be used to enforce policy (e.g. the “FN” must be translated). Authors can save work if not yet ready to move on.
 - Ability to submit translations for review when editing is complete.
 - Ability for a reviewer to assign a batch of concepts from a review pool (exactly analogous to the earlier assignment to authors).

- Ability to review an author's work, make changes, and submit the "final" translation (again with the opportunity to use validation checks to provide additional information or control). Reviewers can save work if not yet ready to send on.
 - Ability to mark a translation as "preview ready" which makes it available to other users of the overall reference set tool for viewing and feedback. This action is reserved for reviewers/admins.
 - Ability to mark a translation as "published" and generate corresponding RF2 artifacts.
- Workflow is implemented in an extensible way with a "default" workflow implementation as described above. Dual independent review and other more complicated workflows will not be supported out of the box but could be developed in the future.
- Enhanced workflow to achieve the stated SNOMED CT Translation Workflow (<http://www.ihtsdo.org/resource/resource/110>)
- Persistence of translation content will also be managed by a handler that will have a default implementation that communicates with the IHTSDO terminology server (to the extent that its capabilities support the needs). If it turns out to be more desirable at this time to use local storage, a local storage option will be employed as well. When storing translated content, a binding of descriptions to the source descriptions will be maintained (where individual descriptions are specifically translated) – otherwise a simple concept connection will be maintained.
- Support for lookups of audit trail and history of translations. This will support the ability to see which authors edited which translations and which reviewers reviewed them. It will also provide a means to see the state of a translation at any point in the past (for example the prior publication state or a prior review state).
- Support the ability to compute the concept and description changes when transitioning from one version of SNOMED CT® to another. This will be done by using effective times to determine the complete set of descriptions and concepts changed in the updated delta. This will support a feature that allows reviewers and admins to see the scope of content that may need to be revisited for translation based on what changed. The concepts involved at this stage can then be put into the workflow (either for re-authoring or simple review).
- Spell checking capabilities based on Lucene and a custom word dictionary provided for the translation project. Spell checking can be handled by validation checks which then provide the author or reviewer an opportunity to add new words to the spelling correction dictionary and re-submit the content.
- Translation suggest feature based to use a dynamic translation memory based on phrase-level translations already approved by reviewers on an individual project. This feature will be implemented with an extensible handler whose default behavior will be to store a 1-n map of possible translations for a particular phrase and it will be specific to individual projects. Thus different "teams" can use different memories.
 - The handler can also be extended to implement things like the google translate API if needed.
 - The handler can also be extended to support reuse of phrase-level translations from other projects if desired. Given localization considerations and the desire to be consistent with across the terminology, it is likely that project level choices will remain specific to that project. However, where no "standard" phrase level translation exists yet, it may be useful for projects to borrow candidate translations from other projects.
 - The feature will support partial translation suggestions where the candidate phrase to translate is longer than phrases currently in the memory. For example, there may be a translation in the memory for "renal failure". When the phrase "acute renal failure" is encountered, the extent of what can be looked up in the memory will be returned as an option.
 - The translation suggest feature will function like an "autocomplete", or a drop-down picklist available to the author at the point of data entry.
 - The feature will support authors adding phrase level translations directly to the memory and it will learn from actual use as well.
 - It would be possible with this framework to enable a validation check (if desired) to warn authors during the translation process if they were using a non-standard phrase level translation. For example, if translating "acute renal failure" where the memory contains a phrase level translation for "renal failure" and the author chooses a translation that does not contain the words of the phrase level translation – a warning could be produced (giving the author a chance to normalize the translation before proceeding).
- Support for the ability to annotate a concept translation in a variety of ways, including simple note, document attachment, or URI link.
- Contextualized help icons associated with each "widget" or "feature" of the client application. For the scope of this project, these icons will lead to pages that contain placeholder content in anticipation of a future effort to develop comprehensive documentation.
- Notification mechanism that allows emails to be sent in response to various workflow events. This will be implemented as a default workflow listener that informs an author or reviewer (if user preferences are configured as such) via email when a new batch of work has been assigned.
- Simple reporting of productivity by time period for authors or reviewers on translation projects. This report, available to reviewers and admins, will simply report the number of concepts/descriptions handled by each author for a specified period of time (daily, weekly, monthly, custom). Other, more sophisticated reporting can be considered in the future, or be derived through API calls.
- As with reference set tooling, this tool will integrate with IHTSDO component identifier service to manage identifier assignment.