OCL Subscription Module (Design Page)

Background

Historically, the The MVP-CIEL Concept Dictionary has been delivered to the community through Dropbox and SQL scripts that require implementations to overwrite their existing dictionary. The process of applying updates is manual, cumbersome, and does not allow for local concepts to exist with CIEL concepts. Open Concept Lab has been an invaluable resource for sharing CIEL and other community vocabularies as well as the mappings between them. As the OCL team is working on a “2.0” vision of the open concept lab that is much more GitHub-like in how dictionaries & collections of concepts are managed, KenyaEMR is looking for a better and more sustainable solution for using the CIEL dictionary and applying changes over time.

Project Champions

- Andrew Kanter as Meta-Data and Terminology Lead
- user-4e734 representing priorities & strategy for OCL
- Steven Wanyee representing priorities & strategy for KenyaEMR

Objectives

Milestone 1 - Let KenyaEMR’s Dictionary server subscribe to CIEL via OCL, while also creating local concepts

Requirements and Assumptions:

1. An OCL service will exist in the cloud, that supports fetching a (large) set of concepts through a web service
   a. It has a built-in JSON format. We may request a specific format.
2. There will be one Kenya concept server on the internet.
   a. Dealing with multiple clients is out of scope for this milestone
   b. Pushing the merged Kenya dictionary to hundreds of Kenya servers is out of scope
3. Implementation can subscribe to one concept sources (initially, Kenya will subscribe to all of CIEL)
   a. Eventually we need to support subscribing to a subset exposed by OCL
   b. Eventually we need to support subscriptions to multiple sources
4. Kenya should be able to download monthly updates from CIEL
5. Implementation can also create local concepts
   a. We must assume that these may have conflicts (e.g. duplicate names)
6. Assumption: implementation should not edit concepts that have been downloaded from a subscription
   a. If they do, it is fair for us to overwrite these changes next download
7. Implementation of OpenMRS must be able to access OCL over the network to update the dictionary. There will not be a way to get updates using e.g. a zipped file.

Current status:

The Open Concept Lab module has been released as 2.0-beta and is currently being tested.

- ✓ Import the full CIEL dictionary into KenyaEMR distribution: import completed in 564 minutes 14 seconds
  - ✓ Fix OCLM-26 - Import prematurely ends with extras object CLOSED Rafal Korytkowski, due to 14 Aug 2015
  - ✓ Fix any reported validation errors. There were 46579 objects updated and 2309 in error. Started a discussion at https://talk.openmrs.org/t/duplicate-concept-names-in-ciel/2779
- ✓ Import the full CIEL dictionary into OpenMRS 2.x distribution, Rafal Korytkowski, due to 17 Aug 2015
  - ✓ Fix any reported validation errors
- ✓ Update OCL to use the latest CIEL dictionary, user-4e734, due to 20 Aug 2015 but not before 18 Aug 2015
- ✓ Import an update for the CIEL dictionary into KenyaEMR, Nicholas Ingosi
- ✓ Import an update for the CIEL dictionary into OpenMRS 2.x, due to 23 Aug 2015
- ✓ Release Open Concept Lab 1.0 module, due to 25 Aug 2015
- ✓ Release of Open Concept Lab 1.1 module
- □ Further testing and deployment of the module in KenyaEMR, due to ??, Nicholas Ingosi and user-4e734

Dev Notes

Design decisions
1. We chose to use OCL API and format provided by OCL instead of requiring from OCL to return format defined by OpenMRS. This implies that on the OpenMRS side we have to create concepts from data provided by OCL and we will be responsible for adjusting the conversion in case any changes to OCL API are made.

2. Concepts from different sources may not conform all OpenMRS validation rules for example they can have duplicate names across sources. Our approach is to import new concepts and updates that are valid against a local database and list the rest for manual fixes. Fixes must be done on the OCL side for concepts coming from OCL and in the local database for local conflicting concepts. It also means that if a valid concept references an invalid concept as an answer, the valid concept will not be imported as it cannot be fully created.

3. There is an open question how to handle cases when the subscription contains concepts with same names from different sources e.g. CIEL:HIV and Kenya:HIV. It was suggested that when creating a subscription you could define that CIEL concepts take precedence over Kenya concepts. The problem is that Kenya:HIV can be referenced by other concepts from the Kenya source and it would have to be replaced by CIEL:HIV. That means the meaning must be exactly the same. They must have (1) the SAME-AS mapping and I think that (2) datatype and (3) class must match as well. This implies that if we discover concepts with same names that do not comply with (1), (2), (3) they will have to be fixed on the OCL side. I'm not sure if it's always possible.

4. Concepts will never be removed by subscribers as they can be referenced already. OCL can only retire them. There must be a retired flag in concept representation and any concept metadata representation (classes, sources, etc.) in OCL.

5. Database IDs assigned to imported items will be different for every subscriber, but will remain the same for every subsequent update.

6. Imported concepts will not be merged in any way with locally created concepts or concepts imported from other sources. If there is a local concept with the same UUID as in OCL, it will be overwritten by the version from OCL when imported. It also means that if there is a local concept with the same name as in OCL, you will not be able to import that concept from OCL unless you retire the local concept (to fix duplicate names).

7. Classes, Datatypes and Sources created locally will be overwritten by those in OCL if they match by name.

8. You can only subscribe to one URL.

Implementation details

OCL API (suggestions)

1. Approach without the RSS feed:
   a. OCL API must expose a subscription URL that lists all (including retired) concepts, sources and mappings in json to be included in the subscription. The assumption here is that we don't have to do extra REST queries to get sources nor mappings and we can find all what is needed in the returned json. The returned JSON must have itemsNumber and updatedBeforeDate fields. The updatedBeforeDate field needs to be set to the value passed in the updatedBeforeDate URL parameter or the date of the server that the json was requested if the parameter was not specified.
   b. The URL needs to accept pagination parameters page=1&per_page=50 (items ordered by uuid for example) and updatedAfterDate=2014-08-13 18:34:23.2314 and updatedBeforeDate=2014-08-13 18:34:23.2314 to display only items updated in the specific time period.
   c. Alternately to b. we could get rid of paging and get a zipped json with all items, which I think should be the preferred solution.

2. Approach with the RSS feed, which as Darius says is more in-line with web conventions:
   a. The RSS feed exposed under a different URL should contain links to updated resources with dates when they were updated so that the client can request the feed and fetch items one by one using resource URLs pointing to specific versions of resources.
   b. For the initial import we still need REST calls described in 1. except for updatedAfterDate support.

3. OCL examples need to be updated to reflect the current state (include missing fields: retired, uuids, ...?)

OpenMRS module

1. The module will support 1.9.5+
2. The module needs to fetch all items from the subscription URL (omitting updatedAfterDate parameter) for a newly added subscription. The date when the items were fetched needs to be stored to query with the updatedAfterDate parameter next time.
3. The module will use a scheduled task to query for updates periodically, importing new items and recording updatedBeforeDate each time.
4. The module can store a subscription URL and updatedBeforeDate in global properties.
5. We will use Jackson library to parse json and RestTemplate from Spring as the REST client.
6. We may need to find a way to handle pagination when saving items. It may happen that an answer to a concept is on a different page than the question. The solution depends on how answers/sets are represented in the concept resource in OCL.
7. It may be inefficient to call OpenMRS services to save changes for ~70k CIEL concepts (it used to slow down MDS imports as well). We may need to investigate how to improve the performance of the save method if we want to use that. The most problematic used to be the concept validation and looking for duplicate names using partial matches (skipping db index), which is hopefully fixed in recent changes (see the recently introduced isConceptNameDuplicate). We may need to back port that to 1.9.x.
8. Reporting validation errors in core needs to be improved. We will need messages to be clear to list items for manual fixes.

User stories:

Adding subscription

1. Administrator enters OCL subscription URL on the Subscription Status page.
2. Module fetches the first page of results (it doesn't pass updatedAfterDate nor updatedBeforeDate parameters) to check if there are any items and notifies the administrator "Updates to dictionary available".
3. Administrator clicks the "Update dictionary" button, which gets disabled and text replaced to "Update in progress" (see the update process)
4. Administrator can see the progress: number of items updated / number of updates and a list of validation errors if any.

Checking subscription status

1. Administrator enters the Subscription Status page.
2. If the subscription is in progress it should see the status "Update in progress" and details.

Manual checking for updates

1. Administrator opens the Subscription Status page and sees the "Check for dictionary updates" button.
2. Administrator clicks the "Check for dictionary updates" button and if there are any updates there's a notification "Updates to dictionary available" and the button changes to "Update dictionary".

Automatic updates

1. Administrator opens the Subscription Status page and can enable checkbox "Update dictionary automatically".
2. It shows "Update dictionary every x days at y", where x is the number of days to set and y is the time when the update should happen.

Update process (in background for paginated json)

1. Module fetches the first page of results and stores the updatedBeforeDate value returned from the server.
2. It saves all concepts, sources and mappings from that URL and updates the progress on the Subscription Status page.
3. Module continues to fetch next pages passing the page and updatedBeforeDate parameters.
4. If there are validation errors module needs to add the issues to the list on the Subscription Status page.
5. If there are validation errors remaining after previous import and not fixed by the new import, the module needs to query for each invalid item individually to see if it can be imported now (i.e. the problem was fixed locally).

Fixing validation errors

1. After invalid items are fixed an administrator can click "Check for dictionary updates and fixes".
2. If any updates are available there's a notification "Updates to dictionary available" and the button changes to "Update dictionary".
3. If no updates are available there's a notification "Updates not available, last time checked: date" and the button "Update dictionary after local fixes".

Questions

1. Classes are not listed as resources in OCL API. They are simple strings in the concept representation. Is that ok from OpenMRS perspective?
2. I don't see answers nor sets in the concept representation in OCL examples. How are concepts of datatype answer or set represented?
3. There was a suggestion to use an RSS feed to query for updates. It adds yet another input format to parse. What are the benefits of having an RSS feed instead of a simple REST call?
4. We base the subscription on querying for concepts changed in the specified period of time. This has the disadvantage that someone can get a concept that is being rearranged and not yet in its final state e.g. concept manager is halfway adding answers to a concept. Is that ok? If not we need to change the approach and expose only published changes. It can be achieved in a few ways, which can be discussed once we decide there's a need for that.
   Andy's answer: The current intent is to have CIEL continue to manage concepts, then publish to OCL and then move to Kenya EMR. Although there might be subsetting going on in OCL, there should not be a lot of temporary concept states that would confuse the ATOM feed.

Resources

1. OCL API https://github.com/OpenConceptLab/oclapi/wiki/concepts
4. Related dev list thread https://groups.google.com/a/openmrs.org/d/topic/dev/kBYnR73QrQo/discussion
7. JIRA https://issues.openmrs.org/browse/OCLM/