Overview: Open Concept Lab (OCL) for OpenMRS

Introduction & Context

One of OpenMRS' strengths is its concept-based data model that allows each implementation to be configured in a suitable way for its own clinical scenario. (Learn more at the Concept Dictionary Basics OpenMRS wiki page.)

This approach also leads to a lot of chaos, as every installation is free to create their own concepts in isolation. Over the years we have created several methods for sharing concepts, but none of these have made it easy to follow best practices, or led to large-scale adoption of concept sharing.

Of note is the curated CIEL concept dictionary, which has been adopted by many OpenMRS sites, but given our current tooling, implementations are faced with an all-or-nothing choice where they can either create their own local concepts, or get updates to CIEL, but not both.

The ideal situation would be if every OpenMRS installation was managing their concepts in a shared service on the web, and the easiest path was for them to reuse existing concepts 90% of the time (either curated ones from preferred sources like CIEL, or crowdsourced ones from some other implementation), and to only create custom concepts when needed.

Open Concept Lab (OCL) promises to be this shared service, but its UI does not allow the typical user to actually carry out the standard workflows they need for proper OpenMRS dictionary management. (For some examples of this see posts 1, 2, and 4 in this thread.)

(See an in-depth description of a proposed workflow for using OCL to manage Bahmni concepts, and the MVP set of workflows. I suggest clicking these links and peeking at the diagrams, but not trying to read the whole posts at this point.)

This project is about:

- Building a new UI from scratch called "OCL for OpenMRS"
- It will be "official" and hosted on OCL servers
- It will be built in a future-looking JS-based technology, and use the OCL REST API
- It would specifically (and only) target the use case of managing OpenMRS dictionaries, so it will be less powerful but easier to use than the traditional OCL UI

The detailed concept and plan is outlined in this living doc(https://docs.google.com/document/d/1aQq3GowirojKG4Djvqn9RdAT9hP_WUw2l3KKyc2ThaM/edit).

Once this project is complete, a typical new OpenMRS implementation would no longer manage concepts via the OpenMRS UI at all. Instead your OpenMRS server would use the Open Concept Lab module to subscribe to a dictionary that you only manage on the cloud through this new application.

What is the Traditional OCL today?

Open Concept Lab runs at https://openconceptlab.org/.

Its backend (https://github.com/OpenConceptLab/oclapi) is written in Python/Django.

The frontend (https://github.com/OpenConceptLab/ocl_web) is a mix of Django web and Angular, but we think of this front end as old code that needs a rewrite on a new tech stack.

In this project we will probably make small changes to the OCL back end, but we will not touch the front end (except that we will include some links in our UI to take people to "traditional OCL").

What is the "OCL for OpenMRS" MVP today?

"OCL for OpenMRS" runs at https://openmrs.qa.openconceptlab.org

OCL's backend (https://github.com/OpenConceptLab/oclapi) is written in Python/Django and exposes a REST API that is used by the OCL for OpenMRS client. Current deployment is at https://api.demo.openconceptlab.org/.

The "OCL for OpenMRS" frontend (https://github.com/openmrs/openmrs-ocl-client/) is a mix of React and Redux. Current deployment is at https://openmrs.demo.openconceptlab.org/home.

What is a "Dictionary" and how does this compare to OCL's existing domain model?

We are trying to simplify things as much as possible for an entry-level OpenMRS user who just wants to manage a dictionary, and doesn't really know anything about concept management, or OCL.

OCL's domain model doesn't actually have anything called a "Dictionary". It has a "repository" which can be either a "source" or a "collection".

We would introduce the idea of a "Dictionary" for this application, and it represents the main thing than an OpenMRS implementer would think about, i.e. it's where you both create your own concepts and include other people's concepts.

In the OCL domain, the "XYZ Dictionary" is represented by:

1. A source called "XYZ Dictionary Custom Concepts". (This could be optional, created on demand the first time that someone creates a custom concept.)

2. A collection called “XYZ Dictionary”. This automatically includes all of the custom concepts from the source (not natively supported by OCL so we need custom code for this), and all concepts that the user manually adds.

We would likely add a REST API to OCL for this domain object, following the “backends for frontends” pattern). See the proposed REST API, and related Talk thread.

Note that in OCL, adding a concept and its mappings is independent, but in our application we will hide that from the user: when you add/remove a concept in the Dictionary, all mappings of that concept are added/removed too.

Relationships between Concepts

In OpenMRS there are strong relationships between concept (Question/Answer, and Set/Set Member), whereas in the OCL API these are treated as mappings, which are weaker.

In our new UI we should treat these as stronger relationships. Specifically, if I add the “What is your favorite color” question to my dictionary, it should automatically also add the answers “blue” and “green”.

Starting From an Existing Dictionary

When you create a new dictionary by copying an existing one, this should:

1. Add a reference to each custom concept from the other dictionary
2. Add all references that are in the other dictionary

Resources

User Guide: Open Concept Lab (OCL) for OpenMRS

Road map: Open Concept Lab (OCL) for OpenMRS

Testing and Continuous Integration Setup for the OCL Client

OCL Client Development Discussion Channel

OCL Client User Feedback Discussion Channel

OCL Development Sprints

JIRA Board

Contributing to OCL for OpenMRS