Building a perfect special-purpose healthcare data model: learning from and assessing OMOP

Vojtech Huser

EPAM RWE/Odysseus

Background

Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM) is a leading Real World Data (RWD) representation format. We collaborated on development of an oncology-specific and purpose-specific[1] and during this development, we assessed how OMOP model captures specific data elements. Consistently adhering to a naming convention for any model may increase the model adoption and facilitate easier model because model users may rely on consistent application of conventions when using many of the model data elements (or table names and column names). In this context, we assessed OMOP model's naming consistency and modelling consistency.

Methods

In the initial step, we obtained a list of all tables and columns in OMOP model. We used a *tag* to group certain related columns to each other. We next defined a higher level *constructs* (above the column name level) and used a tag identical to the title of the construct to group related columns. For example, for construct of ethnicity, we created a tag 'ethnicity' and tagged with it columns of ethnicity_concept_id , ethnicity_source_concept_id, and ethnicity_source_value. For some contructs, the tag also included the table name and dot separator (e.g., drug_exposure.drug)

In building the purpose specific model, we found it helpful to also classify data elements into element categories. We wanted the model to be consistent in using and naming these categories across model entities (e.g., medication, observation, patient, order). Because of this need during our model design and trying to learn from OMOP as much as possible, we considered OMOP data elements and classified table columns into categories. We defined the following categories (with example(s) listed in parentheses): standardized identifier (person_id), source value (ethnicity_source_value), standardized concept (ethnicity_concept_id), source concept (ethnicity_source_concept_id).

We next analyzed the tags and classified constructs (e.g., ethnicity, into several categories based on how many columns were tagged to the construct.

Finally, we found the OMOP concept of 'type' column in most OMOP tables very useful. The type concept idea is used throughout OMOP and provides unique analytical abilities. In our model, we use the term *entity* where OMOP would use a table. E.g., patient entity would be counterpart of OMOP person table. We looked at naming consistency in OMOP for the type column paradigm. In our model, we eventually chose the data element of typeOrigin for OMOP type columns/paradigm.

Results

The tags and categorizations are available at project repository at github.com/informaticsrepo/omop-analysis.

The following construct types were identified:

- A *triple construct* is a construct that in the model has a triple representation and the count of tagged columns is 3. One column is of category standardized concept column category (x_concept_id) and two columns for source value (x_source_concept_id and x_source_value). For example, ethnicity is an example of a triple construct.
- A *double construc*t is a construct that in the model has a double representation and the count of tagged columns is 2. For example, care_site table construct of place_of_service has two tagged columns of place_of_service_concept_id, place_of_service_source_value.

Another way of looking at the triple and double constructs is how many "sibling columns" any given column may have. By siblings columns we mean columns that are somewhat related or linked to the column at hand.

Double constructs further separate into several subtypes. One such subtype is *concept-value double constructs.* Such constructs have columns for standardized concept (x_concept_id) and source value (x_source_value columns). For example, care.site.place_of_service is a concept-value double construct. Concept-source

OMOP model uses triple constructs approach for some data where external terminology may exist and complex modelling is needed (e.g., person.gender, measurement.measurement). For other constructs, a simpler, double construct approach is implemented (e.g., site.place_of_service). Note that for processing data falling under the double construct approach (e.g., place of service), mapping from source value to standardized concept has less model setup infrastructure (no terminology layer for source concept) compared with triple constructs. That is because, triple constructs have the same two columns as double construts but have an additional column for source concept (x_source_concept_id).

Another subtype of double constructs are *identifier-identifier double constructs*. E.g., person table construct of person_identifier with has two tagged columns of person_id and person_source_value and both are of category person-identifier.
The poster (and project repository) will include full overview of how many triple and double constructs were identified and additional analysis resuts.

In terms of naming consistently we identified few of them. One principle may be to name columns with double constructs using a consistent naming pattern. E.g., same prefix used for related columns. OMOP violates this principle for the note table construct of note_class where 2 identically tagged columns are named note_class_concept_id and note_source_value while the naming convention/principle would expect the column names of: note_class_concept_id and note_class_source_value (or rename the construct and drop the class fragment and use names of note_concept_id and note_source_value). The note table also shows a clash in type paradigm and informatics construct of note type (as in LOINC Document Ontology). The name chose in our purpose-specific model (name typeOrigin instead of type) attempts to avoid this naming clash.

Discussion: In model constructions, it may be beneficial to clearly describe which constructs are of what type or subtype. This need is somewhat implicit in OMOP specification by mere presence of columns that follow a naming convention. Although the accompanying guidance clearly defines most of the model building conventions.

Conclusion

We formalized and named several OMOP implicit conventions that were used applied during OMOP model constructions. OMOP model contains powerful features that may inspire other healthcare standardization efforts. Our work attempts to advance the art of perfect model building[2] by identifying useful modelling paradigms.

References

1. Huser V, van Saandijk S, Korsik V. OMOP model evaluation for representing GENIE oncology research project data. OHDSI Europe Symposium, 2025
2. Abellan A, Burn E, Trinh NTH, Burkard T, Callahan A, Fernández-Bertolín S, Hurley E, Rodriguez C, Segundo E, Morales DR, M E Nordeng H, Duarte-Salles T. Expanding the OMOP Common Data Model to Support Perinatal Research in Network Studies. Pharmacoepidemiol Drug Saf. 2025 Feb;34(2):e70106. doi: 10.1002/pds.70106.