Who Wants To Be A 2Billionaire?
A methodology for migrating from STCM to C/CR

Roger Carlson, Matthew Phad, Samuel Martin

Background

Proposed methodology for moving local code mapping data in the SOURCE_TO_CONCEPT_MAP (STCM) table into the CONCEPT and CONCEPT_RELATIONSHIPS (C/CR) tables in the 2-billionaire range.

The STCM was created in OMOP v4 for the purpose of allowing the mapping of local codes to OMOP Standard Concepts. With OMOP v5, that functionality was transferred to the C/CR tables. However, for backward compatibility reasons, the STCM was not deprecated.

The STCM still remains in wide use for several reasons:

- The *Book of OHDSI* [1] recommends its use.
- It is relatively simple to implement.
- *USAGI* [2] (a standard tool that helps with mapping) works with the STCM format.
- Local mapping can be maintained by a separate team unfamiliar with OMOP.
- There is no standard or recommended way to maintain the C/CR method.
- Moving from the STCM to C/CR can involve a lot of ETL code modification.

There are several reasons to stop using SCTM:

- Codes mapped in STCM are not visible in *ATLAS* [3] and other standard OHDSI tools.
- STCM is not flexible enough to map the more subtle relationships available with C/CR like “Maps To Value”.
- Hierarchies are not supported using STCM.

A full comparison of the pros and cons of each approach can be found in *Mapping Custom Source Codes to Standard Concepts: A Comparison of Two Approaches*[4] by Melanie Philofsky.

The OMOP Convention for creating custom concepts can be found in *Custom Concepts (ohdsi.github.io)* by Maxim Moinat and the Themis Working Group .

Methods

For local codes to be available in Atlas, they must be written to the Concept / Concept Relationship tables. It is important to ensure that no concept_ids are over-written, so custom concepts remain stable. At each run, find the highest number already used and set the Sequence. If the mapping for existing concepts changes the record will be updated in the Concept table. New records will be written to the Concept table above the Sequence.

This process involves 4 steps:

1. **SEQUENCE:** Create a Sequence (e.g., 2B_SEQ) for use as the Concept_id.

    a. if max(concept_id) < 2,000,000,000 then sequence = 2000000000*
    b. if max(concept_id) > 2,000,000,000 then sequence = max(concept_id) + 1
    c. *Note: You can start at a higher number if you wish to reserve some 2billionaire values.

2. **VOCABULARY Table**: Create a row in Vocabulary for distinct values of the Vocabulary_ids in STCM
    a. Snapshot existing 2B data and join to STCM
    b. Delete records from Vocabulary that do not exist in the STCM & “CH_generated”
    c. Ignore records that do match
    d. Insert new records that do not exist in the Vocabulary

- VOCABULARY_ID = STCM.SOURCE_VOCABULARY_ID
- VOCABULARY_NAME = STCM.SOURCE_VOCABULARY_ID
- VOCABULARY_REFERENCE = "CH generated"
- VOCABULARY_VERSION = <current_date of the vocab update>

b. VOCABULARY _CONCEPT_ID = 2B_SEQ

VOCABULARY_ID	VOCABULARY_ NAME	VOCABULARY_ REFERENCE	VOCABULARY_ VERSION	VOCABULARY_ CONCEPT_ID
CH_GENDER	CH_GENDER	CH generated	2024-09-10	2100000122

3. **CONCEPT table**: Create 1 row in Concept for each row in STCM.

a. Snapshot existing 2B data (existing_data CTE)
b. Delete records from CONCEPT that do not exist in the STCM
c. Check for new rows to insert or changed rows to update.
d. Place these in a single CTE and MERGE them into existing CONCEPT table.
e. INSERT new records that do not exist in the in existing_data (i.e., not matched in the MERGE)

- CONCEPT_ID = 2B_SEQ (becomes <assigned concept_id>)
- CONCEPT_NAME = STCM.SOURCE_CODE_DESCRIPTION
- DOMAIN_ID = CONCEPT.DOMAIN_ID
- (STCM.TARGET_CONCEPT_ID >--< CONCEPT.CONCEPT_ID)
- VOCABULARY_ID = STCM.SOURCE_VOCABULARY_ID
- CONCEPT_CLASS_ID = CONCEPT.CONCEPT_CLASS_ID
- (STCM.TARGET_CONCEPT_ID >--< CONCEPT.CONCEPT_ID)
- STANDARD_CONCEPT = NULL
- CONCEPT_CODE = STCM.SOURCE_CODE
- VALID_START_DATE = 1970-01-01
- VALID_END_DATE = 2099-12-31
- INVALID_REASON = NULL

f. UPDATE any rows in existing_data have changed (potential changes that would require an update would be if the target concept in STCM changed). CONCEPT row will only be need updating if the target concept has a different DOMAIN_ID or CONCEPT_CLASS_ID then the existing/previous target concept. Be sure to preserve the existing CONCEPT_ID for the concept, allowing for the row to be updated and not inserted.

CONCEPT_ID	CONCEPT_ NAME	DOMAIN_ID	VOCABULARY_ID	CONCEPT_ CLASS_ID	STANDARD_ CONCEPT	CONCEPT _CODE	VALID_ START_DATE	VALID_ END_DATE	INVALID_ REASON
2100000639	Male	Gender	CH_GENDER	Undefined	[NULL]	2	1970-01-01	2099-12-31	[NULL]
2100001971	Female	Gender	CH_GENDER	Undefined	[NULL]	1	1970-01-01	2099-12-31	[NULL]

4. **CONCEPT_RELATIONSHIP Table** Create 2 records in Concept_Relationship for each record in STCM.

a. Since we do not need to preserve CONCEPT_IDs, we can DELETE all mappings currently in CONCEPT_RELATIONSHIP that originated from our STCM.
b. DELETE rows where VOCABULARY_ID like 'CH_%,(as all our custom vocabularies start with "CH_").
c. We also DELETE rows corresponding to concepts that do not currently exist in CONCEPT.

- This accounts for any concepts that were previously in STCM and mapped into CONCEPT/CONCEPT_RELATIONSHIP, but have since been removed from STCM and CONCEPT.

    - This allows for mappings that have changed without having to MERGE/UPDATE the existing table.

d. INSERT new records that are currently in STCM
    - “Maps to” standard concept.
        - CONCEPT_ID_1 = <assigned concept_id>
        - CONCEPT_ID_2 = STCM.TARGET_CONCEPT_ID
        - RELATIONSHIP_ID = "Maps to"
    - “Mapped from” standard concept.
        - CONCEPT_ID_1 = STCM.TARGET_CONCEPT_ID
        - CONCEPT_ID_2 = <assigned concept_id> S
        - RELATIONSHIP_ID = "Mapped from"

CONCEPT_ID_1	CONCEPT_ID_2	RELATIONSHIP_ID	VALID_START_DATE	VALID_END_DATE	INVALID_REASON
2100001971	8532	Maps to	1970-01-01	2099-12-31	[NULL]
210000639	8507	Maps to	1970-01-01	2099-12-31	[NULL]
8532	2100001971	Maps from	1970-01-01	2099-12-31	[NULL]
8507	2100000639	Maps from	1970-01-01	2099-12-31	[NULL]

Results

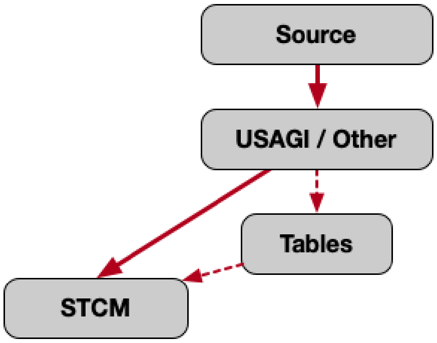
While this mapping can be done directly from the Mapping Source, whether that be USAGI, Athena, Jackalope, or some custom mapping tool, we propose to utilize the Source_To_Concept_Map table as a staging area, which can be effectively used in several use cases.

The process for finding local source codes and mapping them are fully documented in *Book of OHDSI* [1]. Importing the mapping to OMOP, is dependent on the ETL implementation. We intend to have a GitHub repo with the STCM→C/CR code available by the 2024 Symposium.

Use Case 1: New to OMOP: Network Study

An institution may be looking to build an OMOP instance for a specific network study only. They may never use the standard OMOP tool stack at all. The Source_To_Concept_Map (STCM) is by far the simplest and fastest method.

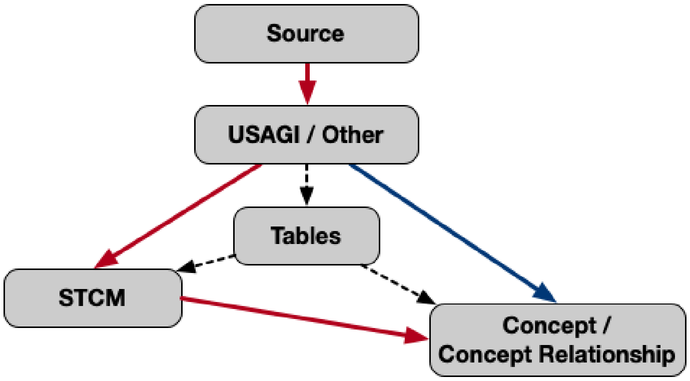
1. Source Codes
    - SQL frequencies
    - CSV files
2. Code Mapping
    - USAGI
    - Other: Athena, Jackalope, Manual
3. Import
    - CSV to separate tables (optional) and/or
    - Write to STCM



Use Case 2: New to OMOP: Enterprise-Wide Use

An institution may be interested in OMOP for enterprise-wide use including implementing the full OMOP tool stack. They may use STCM in the early phase ETL development but need C/CR to display data in Atlas.

1. Source Codes
   - SQL frequencies
   - CSV files
2. Code Mapping
   - USAGI
   - Other: Athena, Jackalope, Manual
3. Import
   - CSV to separate tables (optional) and/or
   - Write to STCM
4. Write from STCM to C/CR

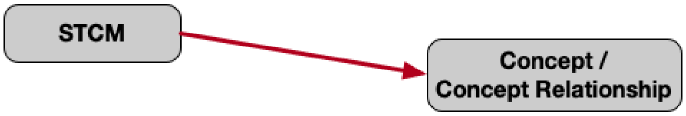


Use Case 3: Expanding Single Purpose OMOP to Enterprise

An institution may have an OMOP instance with ETL built using the STCM (as in Use Case 1) but need to use Atlas and the other OMOP tools. Re-writing their ETL may present a hurdle. They can leave their ETL in STCM, but also write 2billionaire values to C/CR.

3. Write to STCM

4. Write from STCM to C/CR

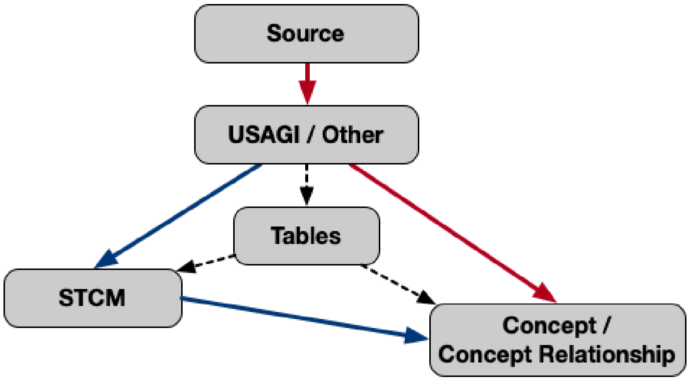


Use Case 4: Complex Mapping and/or Transitioning ETL from STCM to full C/CR

An institution may require complex mappings like "Maps to Value". These mappings can still be done directly to the C/CR, without the use of STCM, while still utilizing STCM for normal mapping. This can also be used to slowly transition ETL from STCM to C/CR.

1. Source Codes
   - SQL frequencies
   - CSV files
2. Code Mapping
   - USAGI
   - Other: Athena, Jackalope, Manual
3. Import
   - CSV to separate tables (optional)

4. Gradually convert ETL from STCM to C/CR



Conclusion

The use of the Source_To_Concept_Map table for local mappings is widely practiced even though more modern alternatives like Concept / Concept Relationship are available. The proposed methodology can be implemented in a variety of OMOP use cases from a simple one-off network study to a full-range enterprise-wide OMOP stack implementation. It can also facilitate an organization's move from one use case to another.

References/Citations

1. The Book Of OHDSI
   https://ohdsi.github.io/TheBookOfOhdsi/
2. The Book Of OHDSI: 6.3.1 Usagi,
   https://ohdsi.github.io/TheBookOfOhdsi/ExtractTransformLoad.html#usagi
3. The Book of OHDSI: 8.3 ATLAS
   https://ohdsi.github.io/TheBookOfOhdsi/OhdsiAnalyticsTools.html#atlas
4. Custom Concepts (ohdsi.github.io) by Maxim Moinat and the Themis Working Group
   https://ohdsi.github.io/CommonDataModel/customConcepts.html
5. Mapping Custom Source Codes to Standard Concepts: A Comparison of Two Approaches, Melanie Philofsky, RN, MS
   https://www.ohdsi.org/wp-content/uploads/2020/10/Melanie-Philofsky-Philofsky-Mapping-Source-Codes-Poster.pdf
6. Code available on GitHub: OHDSI/HealthSystems: Health Systems Interest Group