Analysis of Unit Tests for Source to Target Extract-Transform-Load Code: MIMIC Use Case and Generalization

Vojtech Huser, MD, PhD

Lister Hill National Center for Biomedical Communications, National Library of Medicine, National Institute of Health, Bethesda, MD, USA

Abstract

With increasing adoption of Common Data Models (CDMs), a growing number of healthcare data sources are being converted into a CDM. Also growing is the spectrum of software tools supporting this conversion. We analyzed 35 unit tests for Extract-Transform-Load (ETL) code that converts Medical Information Mart for Intensive Care (MIMIC) dataset into OMOP. We generalized the findings into a set of ETL unit testing lessons learned and conventions applicable to other data sources.

Introduction

With increasing adoption of common data models (CDMs), such as Observational Medical Outcomes Partnership (OMOP), a growing number of healthcare data sources are being converted into a CDM. Also growing is the spectrum of software tools supporting this conversion, such as those developed by the Observational Health Data Sciences and Informatics (OHDSI) community or other CDM-centered communities. Our work presented here focuses on unit testing for Extract Transform Load (ETL) code and a methodological and technical framework for such validation. This work is inspired by a 2020 project that plans to improve the exiting community developed ETL code for Medical Information Mart for Intensive Care (MIMIC) dataset into OMOP model. However, we also try to generalize the ETL unit testing analysis of this use case to future conversions of source data to OMOP target data.

Materials and Methods

Materials: MIMIC-to-OMOP ETL code developed in 2018 (available at https://github.com/MIT-LCP/mimic-omop) uses pgTAP framework that is PostgreSQL-specific. Unit testing is structured by OMOP table and is defined in SQL files named "check_etl.sql". Understandably, unit tests are defined for some OMOP tables (e.g., standardized clinical data tables) while other OMOP tables have no unit tests defined (e.g., CARE_SITE, vocabulary tables and some derived era tables).

Methods: We reviewed and classified existing unit tests by type. We identified MIMIC unit tests that could possibly be added to OHDSI data quality tools (e.g., Data Quality Dashboard [DQD] or Achilles). To enrich our main MIMIC use case analysis, we also studied existing unit test framework within Rabit in a Hat tool (RIAH) and tests for Healthcare Cost and Utilization Project dataset (HCUP) in that framework (created in 2016).[1] When considering possible target data errors[2], we defined the following types: (1) *Incomplete ETL execution error* (target data are incorrect or incomplete due to portion of ETL not being executed as expected); (2) *omission ETL error* (target data are incorrect or incomplete due omission of conversion of portion of data by ETL developers, such as single omitted source event type); and (3) *mapping ETL error* (target data are incorrect or incomplete due to incorrect or missing mapping used by the ETL).

Results

We present results of analysis of phase 1 set of OMOP tables (PERSON, VISIT_OCCURRENCE, MEASUREMENT, DRUG_EXPOSURE, CONDITION_OCURRENCE, and PROCEDURE_OCCURRENCE). The project repository at https://github.com/vojtechhuser/project/tree/master/ETLValidation contains spreadsheet that lists all analyzed MIMIC unit tests as well as additional files and results. In phase 1, we analyzed 34 unit tests. Six unit tests were identical to an existing Achilles analysis. In terms of validation method, the analyzed tests either expected a result value of zero (11 tests; 31.4% of 35 total tests; evaluation whether count of rows with specified WHERE clause criteria is zero) or the two compared queries, on source and target data sorted in a corresponding fashion, had to produce identical table outputs. Our classification of tests consisted of the following categories: distinct count evaluation, source value evaluation, general evaluation, standard concept evaluation, primary key evaluation, and undocumented local concepts evaluation (see project repository for details). In comparison to HCUP RIAH unit tests (where majority of validations consider a single source data row and its converted target form), all MIMIC unit tests spanned the whole OMOP target table or a significant subset of rows (e.g., all rows of particular x_type_concept_id). Such approach may possibly provide a wider validation coverage and detect ETL omission errors.

Generalization to any ETL: MIMIC unit testing code included valuable validation patterns that were applied to multiple tables (e.g., check for standard concepts in CONDITION_OCCURRENCE, MEASUREMENT and DRUG_EXPOSURE) but not applied in general across all tables. Such patterns should be applied on all applicable tables. Another, more important generalization, was greater use of Achilles for ETL Validation. This saves development resources since only one SQL unit testing code (for source data) must be developed that matches an existing Achilles measure. Although, Achilles measure results are not currently ordered. Order logic is necessary for proper source-target match evaluation. In terms of framework, MIMIC ETL is providing X_source_concept_id (always creating *local concept_id's* [in 2 billion range]) for most tables and includes unit test that check that no local concepts are left undocumented in the CONCEPT table (and presumably include an unambiguous concept_name descriptions). Adopting this rule for any ETL allows development of OHDSI tools evaluating mapping accuracy and detecting mapping errors. See project repository for additional findings, lessons learned and possible future work (broken down by existing OHDSI tool).

Implication for Data Quality Tools: We identified two tests that can be added as a new analysis into Achilles. A knowledge base that would establish an "ORDER BY" logic for selected Achilles analyses that useful for ETL validation could also possibly be added.

Mapping validations: The goal for the v2.0 MIMIC ETL (in 2020) is to extend the unit testing to cover mapping errors (how accurate is the mapping of source values to standardized concept_id's). For terminologies included in the OMOP Vocabulary (e.g., dm+d UK drug terminology), OHDSI community can point out inaccurate mappings (and has a history of uncovering mapping errors this way). For local concepts, a framework (based possibly on Natural Language Processing [NLP] methods) is not fully developed in current OHDS tool infrastructure. Although, on study level, StudyDiagnostics package offers some related functionality. Public exposure of mapping (if local data dictionary can be made public) would allow for similar community-based validation for significantly utilized datasets.

Conclusion and Discussion

We analyzed ETL unit testing and formulated generalizable lessons learned for existing or future new OHDSI tools. Our analysis is limited to a single main MIMIC use case (with a limited comparison to HCUP unit tests in RIAH tool). Second limitation is only theoretical analysis (not based on any existing unit tests) of mapping accuracy evaluation.

References

1. Help documentation for Rabit in a Hat testing framework. Available at http://ohdsi.github.io/WhiteRabbit/riah_test_framework.html. [Accessed: July 8,2020]
2. Homayouni H. Master's Thesis: An approach for testing the extract-transfer-load process in data warehouse systems. Fall 2017. Colorado State University. Available at https://www.cs.colostate.edu/etl/papers/Thesis.pdf [Accessed: July 8, 2020]

This research was supported by the Intramural Research Program of the National Institutes of Health/National Library of Medicine/Lister Hill National Center for Biomedical Communications.