OMOP-to-BULK FHIR: A tool to convert population level clinical data into standardized FHIR batch data

Andrey Soares[1], Shahim Essaid[2], Michael G. Kahn[2]
[1] Department of Medicine, University of Colorado Anschutz Medical Campus, [2] Department of Biomedical Informatics, University of Colorado Anschutz Medical Campus

Background

This report describes a tool for converting population level clinical data from OMOP CDM into a Bulk FHIR standard developed for the Multi-State EHR-Based Network for Disease Surveillance (MENDS) project[1], which aims to support public health chronic disease monitoring (hypertension, smoking, statin use, diabetes and obesity). The tool developed covers the data processing pipeline from electronic health record (EHR) data stored in OMOP CDM to a set of FHIR resources that supports Bulk FHIR functionality. This approach provides an alternative for the MENDS surveillance system to consume data from partners in the Fast Healthcare Interoperability Resources (FHIR)/US Core Implementation Guideline (IG) standard. Data from partners can be imported into the electronic medical record support for the ESP data sharing network[2].

In this software demonstration, we will explain the overall architecture for the OMOP-to-Bulk FHIR tool and walk through the steps to convert data from OMOP CDM into FHIR Resources.

Methods

The OMOP-to-Bulk FHIR tool uses OMOP CDM version 5.3, FHIR R4 version 4.0.1 and US Core IG version 4.0.0 (which supports USCDI version 1). Figure 1 shows the data processing pipeline, which includes steps to extract data from OMOP CDM, edit the Whistle translation rules, convert data into FHIR Resources, validate the resulting FHIR Bundle, and upload the bundle to a FHIR Store. The MENDS project would then execute a Bulk FHIR operation to export data into the local ESP repository for distributed queries.

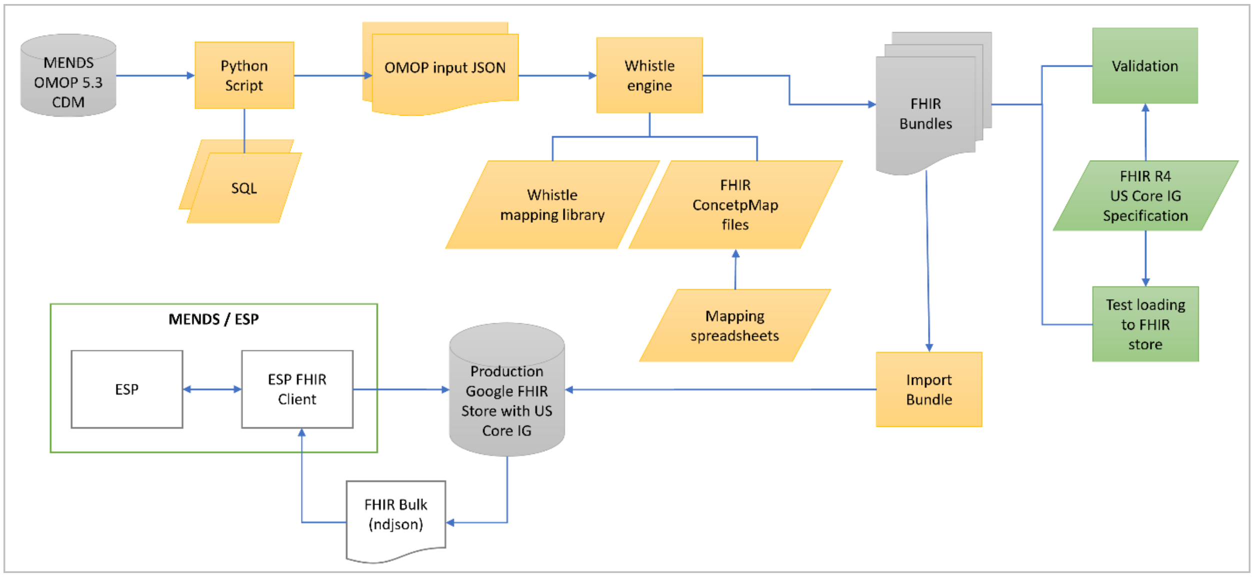
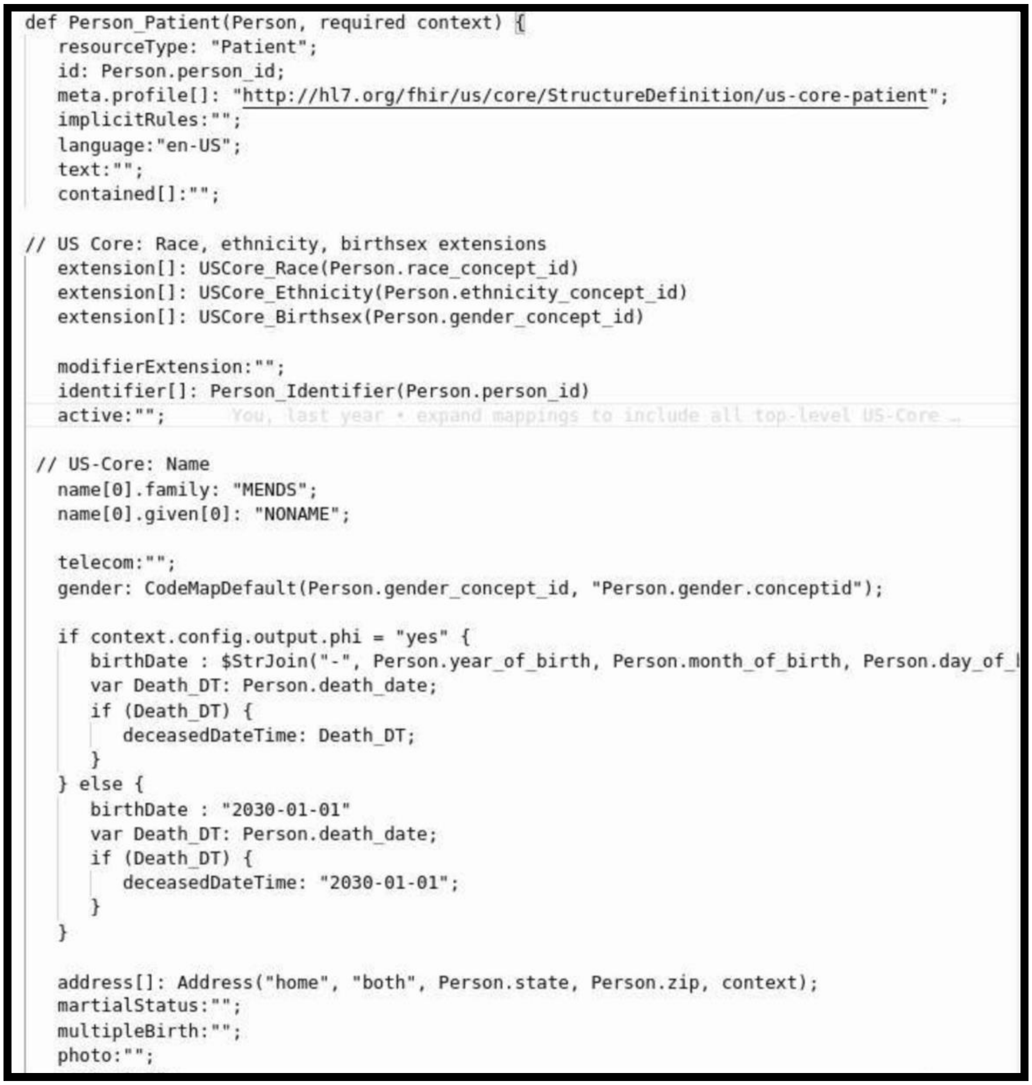


Figure 1: Overall data processing pipeline

The main component supporting the data transformation from one format to another is the Whistle engine[3], developed by Google using the Go Language[4]. Whistle has built-in FHIR-based functions and uses configuration and transformation files to guide the transformation process. A set of Python scripts was created to customized SQL query to retrieve all necessary data for the transformation process. The resulting data is used as input for the Whistle engine. Whistle includes a mapping library to convert data from an OMOP table into a FHIR Resource (Figure 2), and FHIR ConceptMap Resource files with rules to map concepts between terminology systems. For example, the gender "Female" in OMOP (concept_id = 8532 and concept_code = ”F”) is mapped to the code = ”F” and code_system = http://hl7.org/fhir/us/core/ValueSet/birthsex in FHIR.



```
def Person_Patient(Person, required context) {
   resourceType: "Patient";
   id: Person.person_id;
   meta.profile[]: "http://hl7.org/fhir/us/core/StructureDefinition/us-core-patient";
   implicitRules:"";
   language:"en-US";
   text:"";
   contained[]:"";

// US Core: Race, ethnicity, birthsex extensions
   extension[]: USCore_Race(Person.race_concept_id)
   extension[]: USCore_Ethnicity(Person.ethnicity_concept_id)
   extension[]: USCore_Birthsex(Person.gender_concept_id)

   modifierExtension:"";
   identifier[]: Person_Identifier(Person.person_id)
   active:"";      You, last year • expand mappings to include all top-level US-Core …

 // US-Core: Name
   name[0].family: "MENDS";
   name[0].given[0]: "NONAME";

   telecom:"";
   gender: CodeMapDefault(Person.gender_concept_id, "Person.gender.conceptid");

   if context.config.output.phi = "yes" {
      birthDate : $StrJoin("-", Person.year_of_birth, Person.month_of_birth, Person.day_of_
      var Death_DT: Person.death_date;
      if (Death_DT) {
         deceasedDateTime: Death_DT;
      }
   } else {
      birthDate : "2030-01-01"
      var Death_DT: Person.death_date;
      if (Death_DT) {
         deceasedDateTime: "2030-01-01";
      }
   }

   address[]: Address("home", "both", Person.state, Person.zip, context);
   martialStatus:"";
   multipleBirth:"";
   photo:"";
```

Figure 2: Excerpt of a Whistle file with code to convert data from the OMOP Person table into a FHIR Patient Resource

Results

To provide the necessary data for the MENDS project across six domains, we generated FHIR Resources for eight domains. Table 1 shows the mapping of MENDS domains, FHIR Resources and OMOP tables used for this project. Some tables from OMOP were used to map data to different FHIR Resources (ex: the drug_exposure table was mapped to FHIR MedicationAdministration, MedicationRequest, MedicationDispense, Medication and Immunization Resources), and some FHIR Resources were mapped to multiple OMOP tables (ex: the FHIR Patient Resource used data from the OMOP Person, Location and Death tables). In addition, the FHIR Observation Resource can get information from different OMOP tables depending on the target category (ex: OMOP Measurement table is mapped to FHIR Observation Resource with category "laboratory"). To define the mapping from drug_exposure to FHIR resources, we used the drug_type_concept_id field to filter the data. For example, records with the drug_type_concept_id = 32818 (EHR administration record) containing vocabulary_id = "CVX" were mapped to the FHIR Immunization Resource.

Table 1: Domain mapping between MENDS, FHIR and OMOP.

MENDS data domain	FHIR R4 resource(s)	OMOP CDM V5.3 table(s)
Patient	Patient	Person, Location, Death
Encounter	Encounter	Visit_Occurrence
	Condition	Condition_Occurrence
	Coverage	Payer_Plan_Period
	Observation (all)	Observation
Prescription	MedicationAdministration	Drug_Exposure, Drug_Strength
	MedicationDispense	
	MedicationRequest	
	Medication	
Lab Result	Observation (laboratory)	Measurement
Social History	Observation (non-smoking) Observation (smoking)	Observation
Immunization	Immunization	Drug_Exposure, Drug_Strength

The FHIR Bundle generated with this data processing pipeline was validated using the FHIR Validator tool with the US Core IG version 4.0.0. Because of the limitation for internet access from our HIPPA complaint development environment, we evaluated the FHIR Bundles without the use of a terminology server. However, we generate a set of de-identified data to convert and validate the FHIR Bundle outside of the secure environment. This approach allowed us to use the online HL7 Terminology server to evaluate the code systems and codes. The following validation issues were reported by the validator. We interactively identified and corrected the validation issues, and completed the project with less than 1% error rate.

- Error: Code not from code system

- Error: Violations of FHIR invariant rules
- Warning: Code not from value set.
- Warning: No coding from the value set.
- Warning: Label not matching from terminology server
- Info: Unknown extensions
- Info: Unknown code systems

We also evaluated memory and storage needs under different loads, which required adjustment of the code and process. For instance, as the Whistle engine runs in memory, we had to divide the input SQL data into chunks (default 10,000 rows/file) to avoid memory overload and performance issues.

Conclusion

This pilot project created US Core IG conformant FHIR resources using OMOP data from a research data warehouse to provide data for a national chronic disease surveillance system. Enabling OMOP data as a Bulk FHIR interface can make observational clinical data more widely available for research.

References

1. Hohman KH, Martinez AK, Klompas M, Kraus EM, Li W, Carton TW, Cocoros NM, Jackson SL, Karras BT, Wiltz JL, Wall HK. Leveraging Electronic Health Record Data for Timely Chronic Disease Surveillance: The Multi-State EHR-Based Network for Disease Surveillance. J Public Health Manag Pract. 2023 Mar;29(2):162–73.
2. ESPHealth. ESP: Electronic Medical Record Support for Public Health [Internet]. [cited 2023 May 2]. Available from: https://www.esphealth.org
3. Google Healthcare Harmonization Project. Accessed June 12, 2023. Available from: https://github.com/GoogleCloudPlatform/healthcare-data-harmonization
4. Go Language. Accessed June 15, 2023. Available from: https://go.dev