Modeling and Executing Electronic Health Records Driven Phenotyping Algorithms using the NQF Quality Data Model and JBoss® Drools Engine

Dingcheng Li, PhD[1],Cory M Endle[1], Sahana Murthy, MS[1],
Craig Stancl[1], Dale Suesse[1], Davide Sottara, PhD[2], Stanley M. Huff, MD[3],
Christopher G. Chute, MD, DrPH[1] Jyotishman Pathak, PhD[1]
Mayo Clinic, Rochester, MN[1], University of Bologna, Italy[2]
Intermountain Healthcare, Salt Lake City, UT[3]

Abstract

With increasing adoption of electronic health records (EHRs), the need for formal representations for EHR-driven phenotyping algorithms has been recognized for some time. The recently proposed Quality Data Model from the National Quality Forum (NQF) provides an information model and a grammar that is intended to represent data collected during routine clinical care in EHRs as well as the basic logic required to represent the algorithmic criteria for phenotype definitions. The QDM is further aligned with Meaningful Use standards to ensure that the clinical data and algorithmic criteria are represented in a consistent, unambiguous and reproducible manner. However, phenotype definitions represented in QDM, while structured, cannot be executed readily on existing EHRs. Rather, human interpretation, and subsequent implementation is a required step for this process. To address this need, the current study investigates open-source JBoss® Drools rules engine for automatic translation of QDM criteria into rules for execution over EHR data. In particular, using Apache Foundation's Unstructured Information Management Architecture (UIMA) platform, we developed a translator tool for converting QDM defined phenotyping algorithm criteria into executable Drools rules scripts, and demonstrated their execution on real patient data from Mayo Clinic to identify cases for Coronary Artery Disease and Diabetes. To the best of our knowledge, this is the first study illustrating a framework and an approach for executing phenotyping criteria modeled in QDM using the Drools business rules management system.

Introduction

Identification of patient cohorts for conducting clinical and research studies has always been a major bottleneck and time-consuming process. Several studies as well as reports from the FDA have highlighted the issues in delays with subject recruitment and its impact on clinical research and public health [1-3.] To meet this important requirement, increasing attention is being paid recently to leverage electronic health record (EHR) data for cohort identification[4-6]. In particular, with the increasing adoption of EHRs for routine clinical care within the U.S. due to Meaningful Use[7], evaluating the strengths and limitations for secondary use of EHR data has important implications for clinical and translational research, including clinical trials, observational cohorts, outcomes research, and comparative effectiveness research.

In the recent past, several large-scale national and international projects, including eMERGE[8], SHARPn[9], and i2b2[1010101010], are developing tools and technologies for identifying patient cohorts using EHRs. A key component in this process is to define the "pseudocode" in terms of subject inclusion and exclusion criteria comprising primarily semi-structured data fields in the EHRs (e.g., billing and diagnoses codes, procedure codes, laboratory results, and medications). These pseudocodes, commonly referred to as "phenotyping algorithms"[11], also comprise logical operators, and are in general, represented in Microsoft Word and PDF documents as unstructured text. While the algorithm development is a team effort, which includes clinicians, domain experts, and informaticians, informatics and IT experts often operationalize their implementation. Human intermediary between the algorithm and the EHR system for its implementation is required for primarily two main reasons: first, due to the lack of formal representation or specification language used for modeling the phenotyping algorithms, a domain expert has to interpret the algorithmic criteria. And second, due to the disconnect between how the criteria might be specified and how the clinical data is represented within an institution's EHR (e.g., algorithm criteria might use LOINC® codes for lab measurements, whereas the EHR data might be represented using a local or proprietary lab code system), a human interpretation is required to transform the algorithm criteria and logic into a set of queries that can be executed on the EHR data.

To address both these challenges, the current study investigates the Quality Data Model[12] (QDM) proposed by the National Quality Forum (NQF) along with open-source JBoss® Drools[13] rules management system for the modeling and execution of EHR-driven phenotyping algorithms. QDM is an information model and a grammar that is intended to represent data collected during routine clinical care in EHRs as well as the basic logic required to articulate the algorithmic criteria for phenotype definitions. Also, it is aligned with Meaningful Use standards to ensure that the clinical data and algorithmic criteria are represented in a consistent, unambiguous and reproducible manner. The NQF has defined several phenotype definitions (also referred to as eMeasures[14]) many of which are part of Meaningful Use Phase 1, using QDM and have made them publicly available. As indicated in our previous work[15], while eMERGE and SHARPn, are investigating QDM for phenotype definition modeling and representation, the definitions themselves, while structured, cannot be executed readily on existing EHRs. Rather, human interpretation, and subsequent implementation is a required step for this process. To address this need, the proposed study investigates open-source JBoss® Drools rules engine for automatic translation of QDM criteria into rules for execution over EHR data. In particular, using Apache Foundation's Unstructured Information Management Architecture (UIMA)[16] platform, we developed a translator tool for converting QDM defined phenotyping algorithm criteria into executable Drools rules scripts, and demonstrate their execution on real patient data from Mayo Clinic to identify cases for Coronary Artery Disease (NQF74[17]) and Diabetes (NQF64[18]).

Background

A high-level architecture describing the structured rule-based environment for modeling and executing EHR-driven phenotyping algorithms is shown in Figure 1. The environment is divided into three major parts: (1) the phenotyping algorithms complying with QDM, which are developed by the domain experts and clinicians along with informaticians, (2) the executable Drools rules scripts which are converted from (1), and the rules engine that executes the scripts by accessing patient clinical data to generate evaluation results and visualization, (3) the patient data prepared by the domain experts and IT personnel. They are the Extract-Transform-Load of "native" clinical data from the EHR systems into a normalized and standardized Clinical Element Model (CEM)[5] database that is accessed by the JBoss® rules engine for execution. We describe all these aspects of the system in the following sections.

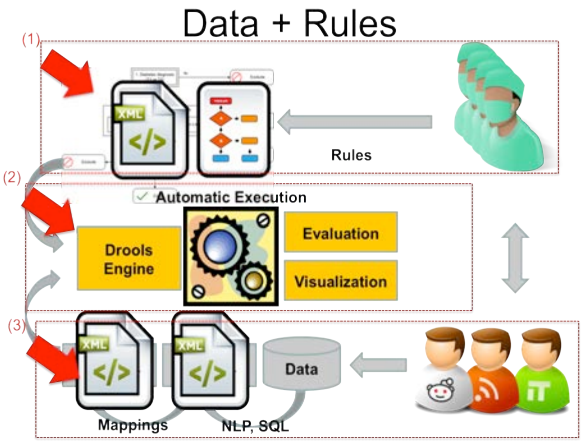


Figure 1 Study Overview (adapted from eMERGE)

NQF Quality Data Model

The QDM is an information model that provides the syntax, grammar and a set of basic logical operators to unambiguously articulate phenotype definition criteria. The model is composed of four parts: category, data type, code list and attributes. Figure 2 shows the basic structure for category and attributes. The category is the highest level definition for QDM elements. Each category involves several states such as *ordered*, *performed*, *documented*, *declined* and etc. and also attributes, such as *timing*, *data flow*, *actors* and *category specific*. The combination of the category and the state forms the primitive data type. For example, medications that have been ordered are represented as MedicationOrdered. QDM allows the measure developer to assign a context in which the category of information is expected to exist. The code list defines the specific instance of the category by assigning a set of values or codes from standardized terminologies and coding systems. The main coding systems used in QDM includes RxNorm[19] for medication, LOINC[20] for lab test results and CPT[21] for billing and insurance, and ICD-9/ICD-10[22]/SNOMED-CT[23] for classifying diseases and a wide range of signs, symptoms, abnormal findings. To facilitate authoring of the phenotype definitions using QDM, the NQF has developed a Web-based tool—the Measure Authoring Tool (MAT)[24] —that provides a way to assess and understand specific healthcare behaviors, and a syntax to relate each QDM elements to other elements within a statement. Three main relationship types included within QDM are: relative timings, operators and functions.

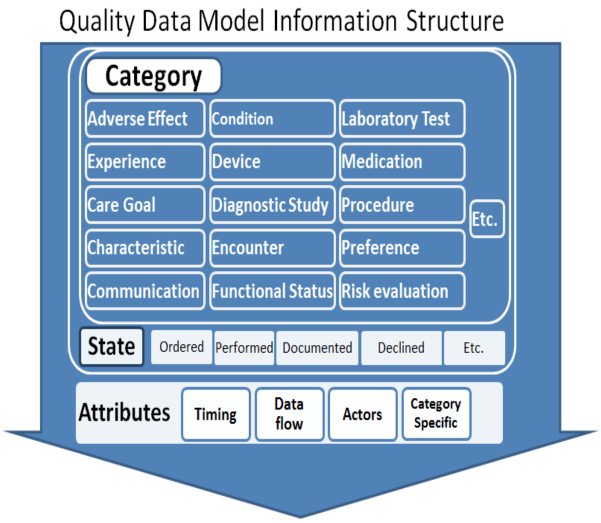


Figure 2 Quality Data Model Information Structure

In addition to above, QDM defines advanced data types under the umbrella of "Population Criteria", and they include: *Initial Patient Population (IPP), Denominator, Numerator, Exclusion, Measure Population* and *Measure Observation*[24]. Each population criteria is composed of a set of declarative statements regulating what values the related primitive data types can take. These statements are organized hierarchically by logical operators, such as AND or OR (detailed examples are discussed in the Evaluation Section). The Initial Patient Population designates the individuals for which a measurement is intended. The Denominator designates the individuals or events for which the expected process and/or outcome should occur. The Numerator designates the interventions and/or outcomes expected for those identified in the denominator and population.

Depending on the purpose of the measure, the Numerator can be a subset of the Denominator (for Proportion measures) or a subset of the Initial Patient Population (for Ratio measures). The Exclusion is displayed in the final eMeasure as Denominator Exclusion, and designates patients who are included in the population and meet initial denominator criteria, but should be excluded from the measure calculation. The Exception is displayed in the final eMeasure as Denominator Exception, and designates those individuals who may be removed from the denominator group. Both Measure Population and Measure Observation are for continuous variable measures only.

As mentioned above, one can specify such criteria via MAT's graphical interface using different QDM constructs and grammar elements as well as specify the requisite code lists from standardized terminologies including ICD-9, ICD-10, CPT, LOINC, RxNorm and SNOMED-CT. In addition to generating a human readable interpretation of the algorithmic criteria (as seen in Figure 2, MAT also produces an XML based representation, which is used by our translator tool for generating the Drools rules scripts (see below for more details).

JBoss® Drools Rules Management System

Drools[13] is an open source, Apache Foundation, community based project that provides an integration platform for the development of hybrid knowledge-based systems. It is developed in Java and has a modular architecture: its core component is a reactive, forward-chaining production rule system, based on an object oriented implementation of the RETE[25] algorithm. On top of this, several extensions are being developed: notably, the engine supports hybrid chaining, temporal reasoning (using Allen's temporal operators[26]), complex event processing[27], business process management (relying on its twin component, the BPMN-2 compliant business process manager jBPM[28]) and some limited form of functional programming.

A Drools knowledge base can be assembled using different knowledge resources (rules, workflows, predictive models, decision tables etc.) and used to instantiate sessions, where the runtime data will be matched against them. In the most common use cases, production rules written in a Drools-specific language (DRL) are used to process facts in the form of java objects, providing a convenient way to integrate the system with existing architectures and/or data models.

The Drools rules can be ordered using salience, assigned to groups, and managed using meta-rules. Groups, in particular, allow to scope and limit the set of rules that can fire at any given moment: only group(s) that are active (focused) allow the execution. Furthermore, due to the tight integration between Drools and jBPM, groups can be layered in a workflow with other tasks. This allows to view the structured processing of a dataset by a rule base as a business process, or dually, to view rule processing tasks as activities within a larger business process. From a practical perspective, a workflow provides a graphical specification of the order of rule execution.

Due to these advantages, our goal is to translate QDM defined algorithms into Drools rules scripts, and demonstrate their execution using EHR data for cohort identification.

Clinical Element Models

As illustrated in Figure 1, a key aspect for execution of the algorithms and Drools rules is an information model for representing patient clinical data. For the SHARPn project, we leverage clinical element models for this task[5, 29]. Developed primarily by SHARPn collaborators at Intermountain Healthcare, CEMs provides a logical model for representing data elements to which clinical data can be normalized using standard terminologies and coding systems. The CEMs are defined using a Constraint Definition Language (CDL) that can be compiled into computable definitions. For example, a CEM can be compiled into XML Schema Definitions (XSDs), Java or C# programming languages classes, or Semantic Web Resources Description Framework (RDF) to provide structured model definitions. Furthermore, the CEMs prescribe that codes from controlled terminologies be used as the values of many of their attributes. This aspect, in particular, enables semantic and syntactic interoperability across multiple clinical information systems.

In Figure 3 , a sample CEM data element for representing Medications Order is shown. This CEM might prescribe that a medication order has attributes of "orderable item", "dose", "route" and so on. It may also dictate that an orderable item's value must be a code in the "orderable item value set", which is a set list controlled codes appropriate to represent orderable medications (e.g., RxNorm codes). The attribute dose represents the "physical quantity" data type, which contains a numeric value and a code for a unit of measure. Instance-level data corresponding to the Medication Order CEM is also shown in Figure 3. The CEMs can be accessed and downloaded from http://sharpn.org.

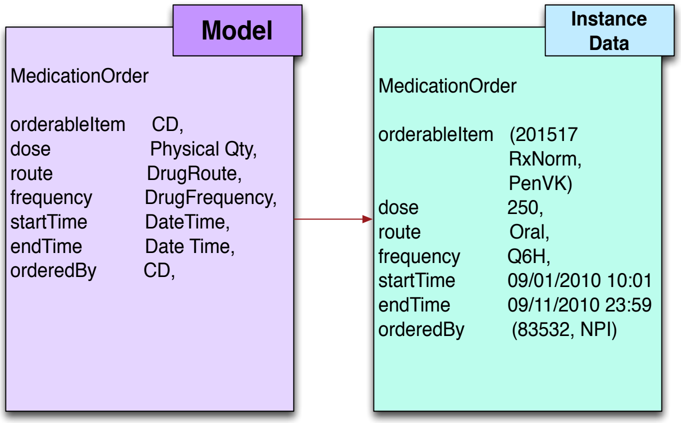


Figure 3 Example CEM for medication order (adapted from Welch et al. [26])

Methods

UMIA-based Translator for Converting QDM Definitions to Drools

As described in Introduction, the goal of the translator is to translate the QDM-based phenotype definitions into a set of executable Drools rules. To achieve this objective, we implemented the translator using the open-source Apache UIMA platform due to the following reasons: firstly, the transformation process leverages the UIMA type system (see details below). Second, UIMA is an industrial strength high-performance data and workflow management system. Consequently, via UIMA, we can decompose the entire translation process into multiple individual sub-components that can be orchestrated within a data flow pipeline. Finally, the use of UIMA platform opens a future possibility for integrating our transformation tool within SHARPn's clinical natural language processing environment—cTAKES[30]—which is also based on UIMA. In what follows, we discuss the four major components and processing steps involved in implementing the QDM to Drools translator.

Figure 4 shows the overall architecture of the QDM to Drools translator pipeline system. Once the eMeasure and phenotype definitions are modeled in NQF's Measure Authoring Toolkit (MAT), in addition to the human readable representation of the phenotype definition criteria (for example, as shown in Figure 6), the MAT also generates an XML and Microsoft Excel files comprising the structured definition criteria conformant to QDM, and the list of terminology codes and value sets used in defining the phenotype definition, respectively. (In Figure 4, these informational artifacts are represented as yellow hexagons). Both the XML (containing the phenotype definition) and Excel (containing the list of terminology codes relevant to the phenotype definition) are required input for the translator system.

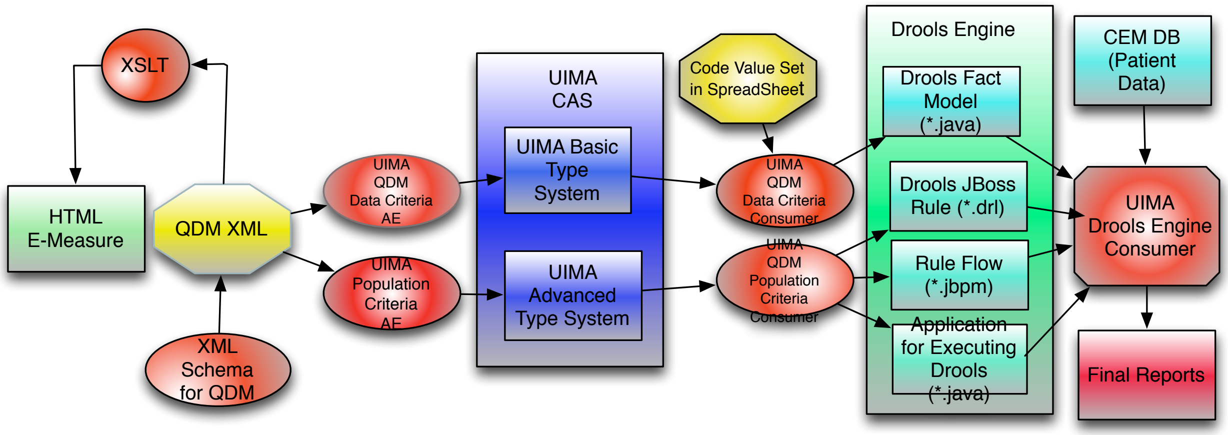


Figure 4 Overall architecture for QDM to Drools translator system

As noted above, our implementation of the translator is based on UIMA that provides a robust type system—a declarative definition of an object model. This type system serves two main purposes: (a) to define the kinds of meta-data that can be stored in the UIMA Common Analysis Structure (CAS), which is the XML-based data structure to store analyzed information; and (b) to support the description of the behavior of a process module. In particular for our implementation, the meta-data from QDM, including both data criteria and population criteria, are defined as UIMA types. All related attributes are also defined accordingly, such that, each UIMA type carries the semantics of corresponding QDM data types in a declarative fashion. Consequently, the UIMA type system not only allows the representation and storage of the QDM data and population criteria as well as their attributes into CAS, but also provides the flexibility to sequentially process this information in a declarative manner.

In UIMA, a CAS Consumer—a Java object that processes the CAS—receives each CAS after it has been analyzed by the analysis engine. CAS Consumers typically do not update the CAS; they simply extract data from the CAS and persist selected information to aggregate data structures, such as search engine indexes or databases. In our work, we adopt a similar approach and define three CAS Consumers—Data Criteria Consumer, Population Criteria Consumer, and Drools Fire Consumer—for processing the QDM data and population criteria as well as for generating and executing the Drools artifacts. Note that as mentioned earlier, a Drools engine architecture is composed of three or four parts: an object-oriented fact model, a set of rules, a rule flow (optional) and an application tool. The fact model, in our case, is generated by the Data Criteria Consumer by extracting code lists from the QDM value sets, and mapping the value sets to corresponding types within the UIMA type system. The rules, which are in essence declarative "if-then-else" statements to implement QDM data criteria, are processed by the Population Criteria Consumer to operate on top of the fact model. The rule flow is a graphical model to determine the order the rule being fired (namely, which data criteria is fired in our work). It allows the user to visualize and thus control the workflow. The application tool is composed of several Java classes that are responsible for creating the knowledge base, including adding all rules and the rule flow.

Mapping QDM Categories to Clinic Element Models (CEMs) for Query Execution

As described above, the SHARPn project leverages CEMs for consistent and standardized representation of clinical data extracted from patient EHRs. While for this study, we persisted CEM XML instances of patient data in a MySQL database, one could potentially serialize the CEM XML files to other representations, including resource description framework (RDF) as demonstrated in our previous work[1]. Regardless of the persistence layer, in order for QDM-based Drools rules to execute on top of a CEM database with patient data, one has to create mappings between both information models. Fortunately, such a mapping can be rather trivially achieved between QDM categories and CEMs. Table 1 below shows how we mapped the QDM categories to existing CEMs developed within the SHARPn project. For example, the Medication CEM (Figure 3) with the qualifiers, such as dose, route, and frequency, can be mapped to the QDM category called Medication with the same attributes. This allows, SQL queries against the CEM database to be executed to retrieve the relevant information required by given rules The entire execution process is orchestrated by the Drools Engine Consumer, which in essence, compiles the Drools rules based on the phenotype definition and logical criteria specified within the QDM Population Criteria, and provides support for both synchronous and asynchronous processing of CEM-based patient data.

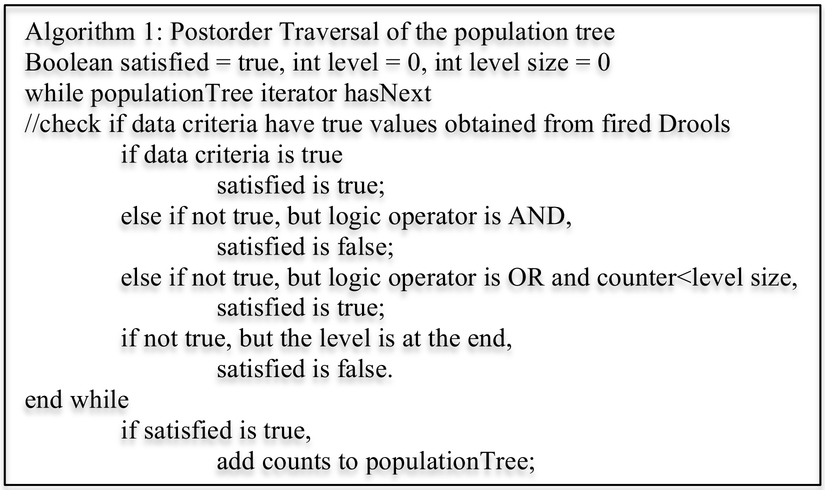
QDM Categories	CEMs
Adverse effect (allergy and non-allergy)	Signs and symptoms
Patient characteristics	Demographics
Condition/Diagnosis/Problem	Problem
Device	Not yet deployed in SHARPn
Encounter	Encounter
Family history	Family History Health Issue
Laboratory test result	Laboratory
Medications	Medication
Procedure performed	Procedure
Physical exam	Basic exam evaluation

Table 1 Mapping between QDM categories and corresponding CEMs

Implementation Evaluation

To evaluate our implementation, we extracted demographics, billing and diagnoses, medications, clinical procedures and lab measurements for 200 patients from Mayo's EHR, and represented the data within the CEM MySQL database. In this initial experiment, we executed two Meaningful Use Phase 1 eMeasures developed by NQF[31]: NQF64 (Diabetes Measure Pair) and NQF74 (Chronic Stable Coronary Artery Disease: Lipid Control) against a CEM-based MySQL database. As shown in the pipeline, once the Drools rules are generated, the Drools Engine Fire Consumer will invoke the Drools engine. In the process, the rules are fired one by one along the rule flow against each patient's corresponding data values. For each patient, a series of Boolean values generated accordingly for each data criteria will be stored into the working memory together with the data values. A key aspect of this entire process is to execute the QDM Population Criteria comprising the Initial Patient Population, Numerator, Denominator, Exceptions and Exclusions with their logical definitions.

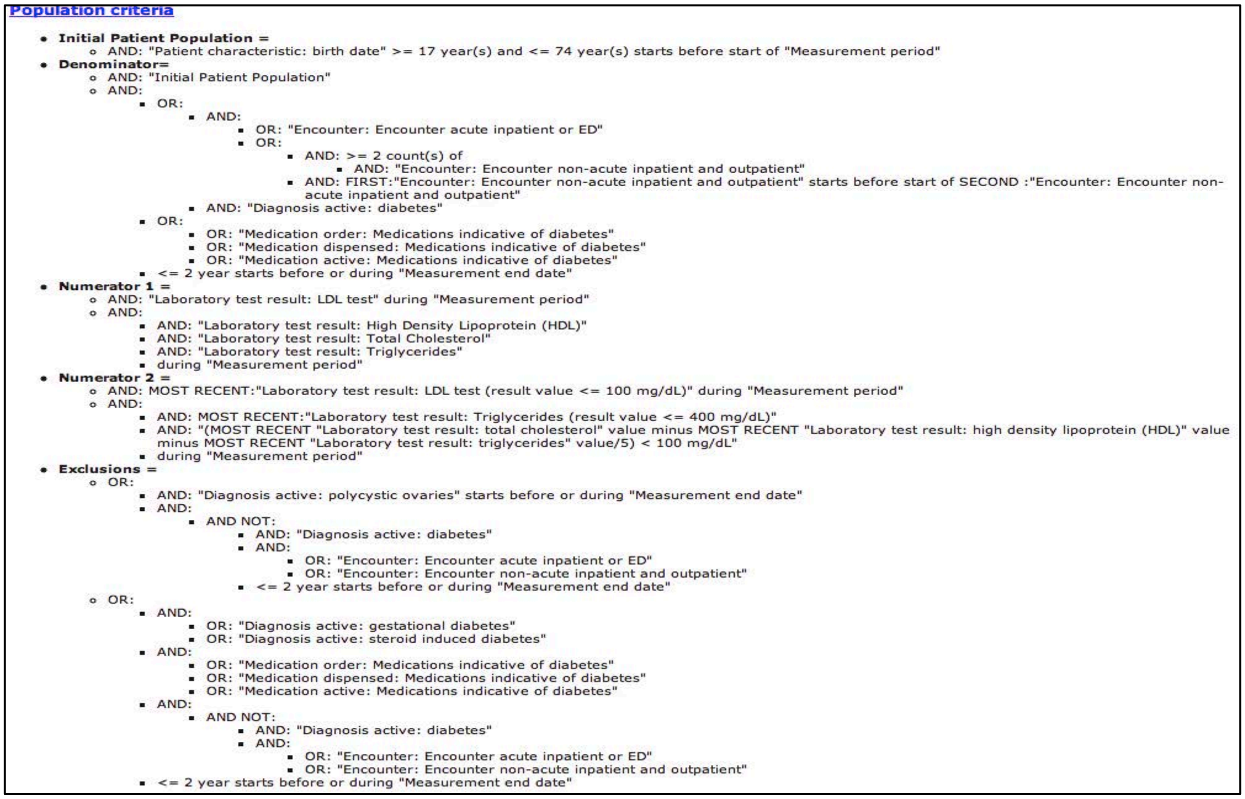
To achieve this, we designed a data structure called population tree to represent the logical structure of the phenotype definition. Each node in a population tree is composed of primitive data types, their logical operators and also their relative locations. The relative location is created to reflect the hierarchical structure of the population criteria. We developed a simple postorder traversal algorithm (Figure 5) that analyzes the Population Criteria, and evaluates if a given patient meets the appropriate criteria for the Numerator, Denominator and Exclusions.



```
Algorithm 1: Postorder Traversal of the population tree
Boolean satisfied = true, int level = 0, int level size = 0
while populationTree iterator hasNext
//check if data criteria have true values obtained from fired Drools
        if data criteria is true
                satisfied is true;
        else if not true, but logic operator is AND,
                satisfied is false;
        else if not true, but logic operator is OR and counter<level size,
                satisfied is true;
        if not true, but the level is at the end,
                satisfied is false.
end while
        if satisfied is true,
                add counts to populationTree;
```

Figure 5 Postorder traversal algorithm for population tree

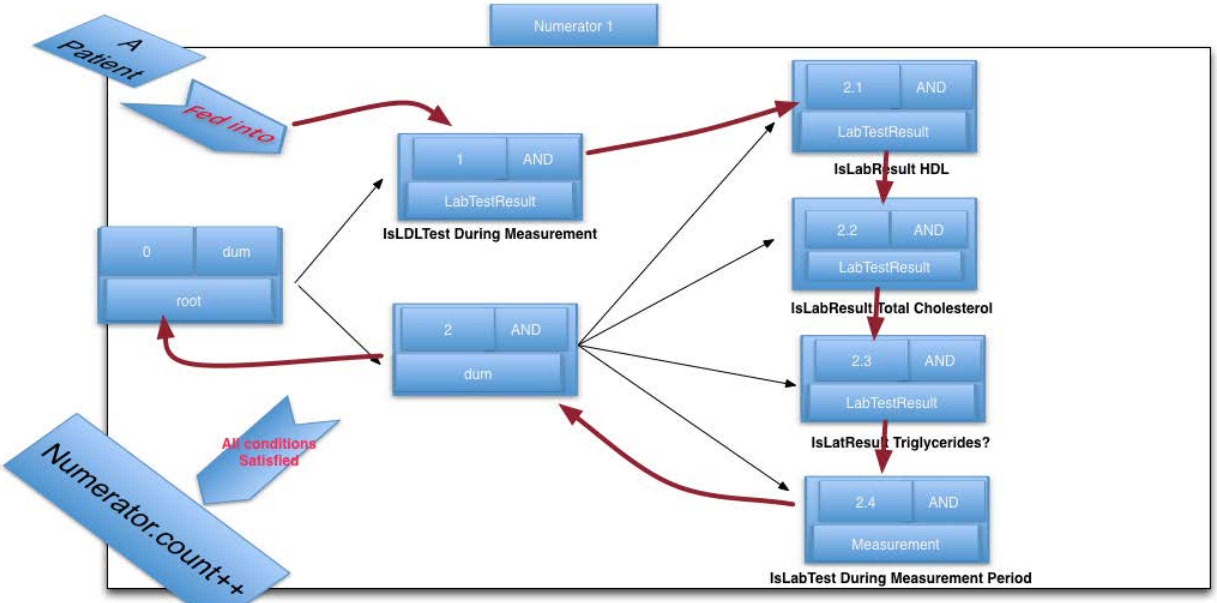
As an example, Figure 6 shows the complete population criteria for NQF64[12] (Diabetes)[31], and the traversal order for one of its Numerator criteria is shown in Figure 7. From left to right, the tree expands from root to leaves. The root is a dummy node without logical operator. There are two branches under the root. The first corresponds to LDL lab test result and the second to an AND node without a data type. Under the second, there are four leaves where their logical levels are from 2.1 to 2.4. After the Drools engine is fired, each patient "walks" through the population tree algorithm as shown in Figure 7. Namely, firstly, a Boolean value, *satisfied* is assigned as true. Next, at each node in the tree, *satisfied* is updated based on the patient's corresponding data value. When the "patient exits" the tree, if *satisfied* is still true, the population count is incremented.



Population criteria

- **Initial Patient Population =**
  - AND: "Patient characteristic: birth date" >= 17 year(s) and <= 74 year(s) starts before start of "Measurement period"
- **Denominator=**
  - AND: "Initial Patient Population"
  - AND:
    - OR:
      - AND:
        - OR: "Encounter: Encounter acute inpatient or ED"
        - OR:
          - AND: >= 2 count(s) of
            - AND: "Encounter: Encounter non-acute inpatient and outpatient"
          - AND: FIRST:"Encounter: Encounter non-acute inpatient and outpatient" starts before start of SECOND :"Encounter: Encounter non-acute inpatient and outpatient"
      - AND: "Diagnosis active: diabetes"
    - OR:
      - OR: "Medication order: Medications indicative of diabetes"
      - OR: "Medication dispensed: Medications indicative of diabetes"
      - OR: "Medication active: Medications indicative of diabetes"
    - <= 2 year starts before or during "Measurement end date"
- **Numerator 1 =**
  - AND: "Laboratory test result: LDL test" during "Measurement period"
  - AND:
    - AND: "Laboratory test result: High Density Lipoprotein (HDL)"
    - AND: "Laboratory test result: Total Cholesterol"
    - AND: "Laboratory test result: Triglycerides"
    - during "Measurement period"
- **Numerator 2 =**
  - AND: MOST RECENT:"Laboratory test result: LDL test (result value <= 100 mg/dL)" during "Measurement period"
  - AND:
    - AND: MOST RECENT:"Laboratory test result: Triglycerides (result value <= 400 mg/dL)"
    - AND: "(MOST RECENT "Laboratory test result: total cholesterol" value minus MOST RECENT "Laboratory test result: high density lipoprotein (HDL)" value minus MOST RECENT "Laboratory test result: triglycerides" value/5) < 100 mg/dL"
    - during "Measurement period"
- **Exclusions =**
  - OR:
    - AND: "Diagnosis active: polycystic ovaries" starts before or during "Measurement end date"
    - AND:
      - AND NOT:
        - AND: "Diagnosis active: diabetes"
        - AND:
          - OR: "Encounter: Encounter acute inpatient or ED"
          - OR: "Encounter: Encounter non-acute inpatient and outpatient"
        - <= 2 year starts before or during "Measurement end date"
  - OR:
    - AND:
      - OR: "Diagnosis active: gestational diabetes"
      - OR: "Diagnosis active: steroid induced diabetes"
    - AND:
      - OR: "Medication order: Medications indicative of diabetes"
      - OR: "Medication dispensed: Medications indicative of diabetes"
      - OR: "Medication active: Medications indicative of diabetes"
    - AND:
      - AND NOT:
        - AND: "Diagnosis active: diabetes"
        - AND:
          - OR: "Encounter: Encounter acute inpatient or ED"
          - OR: "Encounter: Encounter non-acute inpatient and outpatient"
    - <= 2 year starts before or during "Measurement end date"

Figure 6 Population Criteria for NQF64 (Diabetes)



A sample traversal for Numerator 1 of NQF 64 (diabetes). Starting from 1, a patient will walk through Tree Numerator 1 in postorder fashion.

1. At 1, Lab test result will be checked. Since the logical operator is AND, then if the LDL test is done during the measurement period, satisfied is true. Otherwise, satisfied is false and break.
2. Go to 2.1, 2.2 and 2.3, the logical operators are all AND. Hence, check if lab test result includes HDL, total cholesterol and triglycerides during measurement period. If all true, satisfied is true. If one of them is false, satisfied is false and break.
3. Then, the patient walks back up to 2 and 0. No need to check.
4. If satisfied is true, Numerator count increase. Otherwise, continue to loop, and evaluate the next patient instance.

Figure 7 The sample traversal for Numerator of NQF 64

Once all the numerator, denominator and exclusion criteria have been processed, the next step is to calculate the eMeasure proportions defined as Figure 8. by NQF[31]: Similar to our implementation and evaluation of NQF64 eMeasure for Diabetes, we experimented with NQF74 eMeasure for Coronary Artery Disease, although due to space limitations, we exclude the discussion about NQF74.

- Calculate the final denominator by including all the patients that meet the denominator criteria.
- Subtract from the final denominator all patients that do not meet numerator criteria, yet satisfy the exclusion criteria.
- Define the final calculation as follows:
    - For "Proportion" measures, the calculation is the total number of patients satisfying the numerator criteria divided by the final denominator.
- For measures with multiple denominators, repeat the above process for each denominator and report each result separately.
- For measures with multiple patient populations, repeat the above process for each patient population and report each result separately.
- For measures with multiple numerators, calculate each numerator separately within each population using the paired exclusions (where applicable).

Figure 8 Population Calculation Psudocode

Discussion

In this study, we propose and implement an approach for structured representation of phenotype definitions using the NQF Quality Data Model, and its execution on EHR data by using the open-source JBoss Drools business rules management environment. A similar study was conducted by Jung et al.[32] where the authors investigated automatic translation of phenotype definition and criteria represented using HL7 Arden syntax to Drools rules using XSLT stylesheets. The XSLT stylesheets were also used to create a user interface for manual validation of the Drools rules, and eventually create an application to drive the execution of the rules.

There are few related efforts [10, 31,32] that have attempted in the automatic conversion of phenotyping definitions into executable code. Most notably, Jung et al. [32] implemented a platform for expressing clinical decision support logic using the HL7 Arden syntax that can be automatically translated to executable Drools scripts. Unlike our UIMA-based approach, this translation was done using Extensible Stylesheet Language Transformations (XSLTs).

While similar in spirit, we believe our work highlights at least three significant aspects. First, to the best of our knowledge, this is the first study that has attempted to automatically generate executable phenotyping criteria based on standards-based structured representation of the cohort definitions using the NQF QDM. Second, we demonstrate the applicability of open-source, industrial-strength Drools engine to facilitate secondary use of EHR data, including patient cohort identification and clinical quality metrics. Finally, we have successfully integrated UIMA and JBoss Drools engine within a run-time execution environment for orchestration and execution of phenotyping criteria. This opens future possibilities to further exploit the advanced process and/or event management technologies contained in jBPM and Drools Fusion. The latter, in particular, handles CEP (Complex Event Processing, which deals with complex events) and ESP (Event Sequence Processing, which deals with real-time processing of huge volume of events).

However, there are several limitations outstanding issues that warrant further investigation. For example, in this study, we investigated only two NQF eMeasures to evaluate the translation system. While both these eMeasures are complex in nature and represent various elements of the QDM, additional evaluation is required. To this end, we have already started implementing the existing phenotyping algorithms developed within the eMERGE consortia as well as the Meaningful Use Phase 1 eMeasures defined by NQF. Additionally, in this first effort, our focus was on core logical operators (e.g., OR, AND). However, QDM defines a comprehensive list of operators including temporal logic (e.g, NOW, WEEK, CURTIME), mathematical functions (e.g., MEAN, MEDIAN) as well as qualifiers (e.g., FIRST, SECOND, RELATIVE FIRST) that require a much deeper understanding of the underlying semantics, and subsequent implementation within the UIMA type system. Our plan is to incorporate these additional capabilities in future releases of our translation system.

Conclusions and Future Work

In conclusion, we have developed a UIMA-based QDM to Drools translation system. This translator integrates XML parsing of QDM artifacts, as well as Drool scripts and fact model generation and eventually execution of the Drools rules on real patient data represented within clinical element models. We believe we have demonstrated a promising first step towards achieving standards-based executable phenotyping definitions, although a significant amount of future work is required, including further development of the translation system as QDM continues to evolve and is being updated.

As mentioned above, our next step is to expand the framework to all the existing Meaningful Use Phase 1 eMeasures. Additionally, we plan to develop a Web services architecture that can be integrated with graphical and user friendly interfaces for querying and visual report generation. And last, but not the least, in the current implementation with the CEM database, we are limited to information that is contained within the CEM instances of patient data. Hence, the capability to execute free-text queries on clinical notes, for example, is missing. As mentioned earlier, since SHARPn's clinical natural language processing pipeline—cTAKES—is also based on UIMA, one of our objectives is to extend the current implementation to incorporate processing of queries on unstructured and semi-structured clinical text.

Software access. The QDM-to-Drools translator system along with sample eMeasures NQF64 and NQF74, can be downloaded from: http://informatics.mayo.edu/sharp/index.php/Main_Page.

Acknowledgment. This manuscript was made possible by funding from the Strategic Health IT Advanced Research Projects (SHARP) Program (90TR002) administered by the Office of the National Coordinator for Health Information Technology. The authors would like to thank Herman Post and Darin Wilcox for their inputs in implementation of the QDM to Drools translator system. The contents of the manuscript are solely the responsibility of the authors.

References

1. Cui Tao CGP, Thomas A. Oniki, Jyotishman Pathak, Stanley M. Huff, Christopher G. Chute: **An OWL Meta-Ontology for Representing the Clinical Element Model**. *American Medical Informatics Association* 2011:1372-1381.
2. Chung TK KR, Johnson SB: **Reengineering clinical research with informatics**. *Journal of investigative medicine* 2006, **54**(6):327.
3. F S Mair PG, C Shiels, C Roberts, R Angus, J O'Connor, A Haycox and S Capewell: **Recruitment difficulties in a home telecare trial**. *Journal of Telemedicine and Telecare* 2006, **12**:126-128.
4. Abel N. Kho JAP, Peggy L. Peissig, Luke Rasmussen,Katherine M. Newton, Noah Weston, Paul K. Crane, Jyotishman Pathak, Christopher G. Chute, Suzette J. Bielinski, Iftikhar J. Kullo, Rongling Li,Teri A. Manolio, Rex L. Chisholm and Joshua C. Denny: **Electronic Medical Records for Genetic Research: Results of the eMERGE Consortium**. *Science Translational Medicine* 2011, **3**(79):3-79.
5. Chute CG PJ, Savova GK, Bailey KR, Schor MI, Hart LA, Beebe CE, Huff SM: **The SHARPn project on secondary use of Electronic Medical Record data: progress, plans, and possibilities**. In: *American Medical Informatics Association: Oct 22, 2011 2011; Washington DC*. 248-256.
6. Isaac S Kohane SEC, Shawn N Murphy: **A translational engine at the national scale: informatics for integrating biology and the bedside**. *Journal of American Medical and Informatics Association* 2012:181-185.
7. David Blumenthal aMT: **The "Meaningful Use" Regulation for Electronic Health Records**. *The New England Journal of Medicine* 2010:501-504.
8. McCarty CA CR, Chute CG, Kullo I, Jarvik G, Larson EB, Li R, Masys DR, Ritchie MD, Roden DM, Struewing JP, Wold WA, and the eMERGE team: **The eMERGE Network: A consortium of biorepositories linked to electronic medical records data for conducting genomic studies**. *BMC Medical Genomics* 2011, **4**(1):13.
9. Kho AN PJ, Peissig PL, Rasmussen L, Newton KM, Weston N, Crane PK, Pathak J, Chute CG, Bielinski SJ, Kullo IJ, Li R, Manolio TA, Chisholm RL, Denny JC: **Electronic Medical Records for Genetic Research: Results of the eMERGE Consortium**. *Science Translational Medicine* 2011, **3**(79):79re71.

10. **i2b2 - Informatics for Integrating Biology & the Bedside** [https://http://www.i2b2.org/software/ - downloadables]
11. **Library of Phenotype Algorithms** [https://http://www.mc.vanderbilt.edu/victr/dcc/projects/acc/index.php/Library_of_Phenotype_Algorithms]
12. **Quality Data Model** [http://www.qualityforum.org/Projects/h/QDS_Model/Quality_Data_Model.aspx]
13. **Drools - The Business Logic integration Platform** [http://www.jboss.org/drools]
14. **National Quality Forum** [http://www.qualityforum.org/Measures_List.aspx]
15. Mike Conway RLB, David Carrell, Joshua C. Denny, Abel N. Kho, Iftikhar J. Kullo, James G. Linnerman, Jennifer A. Pacheco, Peggy Peissig, Luke Rasmussen, Noah Weston, Christopher G. Chute, Jyotishman Pathak: **Analyzing the Heterogeneity and Complexity of Electronic Health Record Oriented Phenotyping Algorithms**. In: *American Medical Informatics Association: October 22, 2011 2011*. 2011: 274-283.
16. **Unstructured Information Management Applications** [http://uima.apache.org/]
17. **NQF74: Chronic Stable Coronary Artery Disease** [http://www.qualityforum.org/MeasureDetails.aspx?actid=0&SubmissionId=379 - p=8&s=n&so=a]
18. **NQF64 - Diabetes Measure Pair** [http://www.qualityforum.org/MeasureDetails.aspx?actid=0&SubmissionId=1228 - p=7&s=n&so=a]
19. **RxNorm** [http://www.nlm.nih.gov/research/umls/rxnorm/]
20. **LOINC: Logical Observation Identifiers Names and Codes** [http://loinc.org/]
21. **CPT Coding, Billing & Insurance** [http://www.ama-assn.org/ama/pub/physician-resources/solutions-managing-your-practice/coding-billing-insurance/cpt.page]
22. **ICD-9/ICD-10** [http://icd9cm.chrisendres.com/]
23. **SNOMED Clinical Terms** [http://www.nlm.nih.gov/research/umls/Snomed/snomed_main.html]
24. **Measure Authoring Tool Basic User Guide**: the National Quality Forum; 2011.
25. Forgy C: **Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem**. *Artificial Intelligences* 1982, **19**(1):17-37.
26. Allen JF: **An Interval-Based Representation of Temporal Knowledge**. In: *7th International Joint Conference on Artificial Intelligence: 1981*. 221-226.
27. Luckham D: **The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems**: Addison-Wesley Longman Publishing Co., Inc.; 2002.
28. **Java Business Process Management** [http://www.jboss.org/jbpm]
29. Susan Rea JP, Guergana Savova, Thomas A. Oniki, Les Westberg, Calvin E. Beebe, Cui Tao, Craig G. Parker, Peter J. Haug, Stanley M. Huff, Christopher G. Chute: **Building a robust, scalable and standards-driven infrastructure for secondary use of EHR data: The SHARPn project**. *Journal of Biomedical Informatics* 2012.
30. Guergana K Savova JJM, Philip V Ogren,Jiaping Zheng, Sunghwan Sohn, Karin C Kipper-Schuler, Christopher G Chute: **Mayo clinical Text Analysis and Knowledge Extraction System (cTAKES): architecture, component evaluation and applications**. *Journal of American Medical and Informatics Association* 2009, **17**:507-513.
31. **Measure Authoing Tool** [http://www.qualityforum.org/MeasureDetails.aspx?actid=0&SubmissionId=1228]
32. Jung CY, Katherine A Sward, Peter J Haug: **Executing medical logic modules expressed in ArdenML using Drools**. *Journal of American Medical and Informatics Association* 2012.