Distributed Health Data Networks: Implementing a Scalable Query Interface within PopMedNet for Use in Large-Scale Diverse Networks

Jessica Malenfant, MPH

Department of Population Medicine of Harvard Medical School & Harvard Pilgrim Health Care Institute

Health Care Systems Research Network Annual Conference
April 12, 2018
Objective

Demonstrate a new architecture and framework for an extensible point-and-click query interface in PopMedNetTM (PMN) that:

- Addresses challenges in platform and software heterogeneity in PCORnet, the largest PMN network
- Are modularized and can successfully target multiple data models and various technical ecosystems
- Utilize widely adopted standard data exchange formats e.g. JSON, LINQ, Microsoft Entity Framework, and SQL
- Produce consistent and valid results
Background

- PMN powers clinical and observational research through efficient and privacy-preserving methods and technologies
- PMN infrastructure permits investigators to compose and distribute custom queries through a variety of tools
- PMN is a mature platform that is used by 100s of organizations
- PMN is used in several large-scale distributed data networks including: PCORI’s PCORnet and FDA’s Sentinel Initiative
Problems Identified with the Initial MDQ Tool

Legacy Query Composer: Developed for limited use resulting in scalability issues:

- Each query tool was hardcoded for use against a single CDM and database platform
  - The MDPHnet network’s ESP data model and PostgreSQL
  - FDA’s Sentinel System Summary Table data model and MS Access
- Changes required manual and redundant hard-coding
- Queryable terms could not be shared across networks (e.g. if 2 networks wanted to query race data, each query tool needed to be developed separately, even if the field names and value sets were the same)
- Changes required the sites to download a new version of the PMN DataMart Client software in order to respond to a query
End users want a simple query tool interface and workflow

Infrastructure should be re-usable and easily extensible and scalable, limiting CDM-specific coding

Address the heterogeneity of technical environments across the large-scale distributed networks PMN supports

Consider workflows for full request lifecycle including integration points with external systems

Challenge: Develop a One Size Fits All MDQ Tool
Request Cycle

Challenges to Consider:

Primary source data: refresh rates vary across sites, ETL processes may vary

CDM: Could be 1 of many approved CDM versions

RDBMS: Could be 1 of many supported database systems and versions of the RDBMS

Technical environment: DMC is Windows app, data may live in a Linux/Unix & involve manual processes to query data
Tools Developed

Menu-Driven Queries (MDQs):

- PMN interface supports querying terms and stratifications
- Investigators can compose a simple or complex MDQ that includes logical operators “OR”, “AND”, “AND NOT” to define a cohort of interest via a user interface
- Include software-enabled governance to determine what users can query
- Support electronic workflows and embedded analytics
- Include data model adapters that make the MDQs Common Data Model (CDM) aware
- Modular design for sharing queryable terms regardless of data source

Test Case Inserter (TCI):

- Generates databases according to CDM specifications
- Custom program that enables users to easily insert synthetic data into a relational database management system (RDBMS) without requiring the user to have SQL programming skills
- Supports MDQ validation and MDQ prototypes for targeting new data sources
One Size fits Most* MDQ Tool

*Committed to support the most common RDMBS versions used across PCORnet
**Investigator/Coordinating Center**

- **Menu-Driven Query**
- **Distribute Request to Data Partners**
- **Investigator / Analyst Downloads Request Responses from Each Data Partner**
- **Data Analysis**
- **DP #1 Results**
- **DP #N Results**

**Data Partner N**

- **DataMart Client (DMC)**
- **DataMart Admin Review and Run Query**
- **Transfer Request & Response Between Requestor & Data Partner(s)**
- **Data Partner N**
- **CDM v.X.Y**
- **Oracle**
- **Microsoft SQL Server**
- **PostgreSQL**

*DP = Data Partner*
Menu-Driven Query Process

1. Distribute Request to Data Partners
2. Transfer Request & Response Between Requestor & Data Partner(s)
3. DataMart Admin Review and Run Query
4. Data Mart Client (DMC)
5. Investigator / Analyst Downloads Request Responses from Each Data Partner

*DP = Data Partner
MDQ Technical Details and Validation

MDQs are currently enabled to query terms and fields found in many data models. Data partners download the PMN DataMart Client (DMC) application and configure it to match their data model, as well as connect it to their local RDBMS.

The TCI tool generates data sets and databases that match any given Common Data Model (CDM). TCI then inserts the data into a supported RDBMS.

**Data Sources**

- **RDBMS Platforms Tested**
  - Oracle 11, 12
  - Postgres 9.4, 9.5, 9.6

- MDQs are currently developed to match the PCORNet CDM. Non-PCORNet data sources can utilize MDQs if they share concepts e.g. PCORNet uses DX_TYPE and Sentinel uses DX_CODE_TYPE to represent diagnosis code type.

**TCI**

The TCI tool generates data sets and databases that match any given Common Data Model (CDM). TCI then inserts the data into a supported RDBMS.

**Query composed in PMN & distributed to data partners**

- JSON

**Data partner receives query through PMN DMC application**

- JSON

**Query parsed by DMC data model adapter according to CDM**

- LINQ

**Query translated into SQL appropriate for data partner’s RDBMS**

- SQL

**Data partner executes query against local database and uploads results from the DMC**

- JSON

**Results securely return to PMN investigator**
MDQ Use Case

How many patients with high cholesterol and high blood pressure but do not have heart failure?

Use MDQ to find patients of interest. Terms are added to the PMN MDQ interface according to the data model. Terms can be re-purposed for other data models.
Query Interface to Define Cohort

### Criteria Group 1:
Hypertension with visits between 2000-2016

<table>
<thead>
<tr>
<th>Criteria Group: Hypertension</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Group Name</strong></td>
</tr>
<tr>
<td>Hypertension</td>
</tr>
</tbody>
</table>

- **Diagnosis**: Hypertension
- **Code Set**: ICD-9-CM
- **Search Method**: Exact Match
- **Selected Codes**: 4019
- **Observation Period**:
  - Start: 01/01/2000
  - End: 12/31/2016
Query Interface to Define Cohort

Criteria Group 2: AND patients have high cholesterol
Query Interface to Define Cohort

Criteria Group 3: AND patients without heart failure
DataMart Administrator Receives the Query

DataMart Administrator Inbox – locally installed app at each site
DataMart Administrator Reviews Query Details

Administrator can review query input

Request JSON transmitted from the web portal to the DMC can also be viewed by users.
DataMart Administrator Executes the Query and Reviews Results

Once request is run locally, the LINQ generated SQL is also available to the user.

This is the database agnostic query language that is then translated into a specific SQL flavor by the RDBMS service provider.
...and send results back to the requestor if they choose to
Investigator Reviews Site-Specific Results on Web Portal

MDQ Results:
Patients with hypertension diagnosis with visits between 2000-2016
AND patients have high cholesterol ICD-9 diagnosis codes
AND patients without heart failure diagnosis codes
Multiple filters and stratification options have been added to the MDQ tool for several fields including Race, Sex, Observation Period, Diagnosis and Procedure Codes, Height, Weight, Age, etc., more planned.

The PCORnet data adapter has been updated to process queries with the new terms and stratification options.

Testing with the TCI tool has verified that ad hoc data models that share PCORnet CDM fields can use the MDQ out-of-the-box, continuing to explore.
Current Status

- Enhancing automation functionality, including expanding distributed regression analysis functionality
- Preparing to add functionality for users to define index events
- Validation and performance testing is in progress to evaluate how complex queries behave
- Ability to expose the actual SQL to a user prior to running a query is under investigation. The request JSON and the LINQ code are currently available to end users but require manual steps to piece the query languages together, for example:
Current Status

-- PrimaryObservationStart: '10/15/2013 12:00:00 AM' (Type = DateTime2, IsNullable = false)
-- PrimaryObservationEnd: '10/14/2014 12:00:00 AM' (Type = DateTime2)
-- PrimaryObservationStart: '10/15/2013 12:00:00 AM' (Type = DateTime2)
-- PrimaryObservationEnd: '10/14/2014 12:00:00 AM' (Type = DateTime2)
-- CriteriaOneCodeType: '09' (Type = String, Size = 4000)
-- CriteriaOneMinimumAge: '65' (Type = Int32, IsNullable = false)

```
SELECT 1 AS [C1],
[GroupBy1].[X1] AS [SEX],
[GroupBy1].[X2] AS [HISPANIC],
[GroupBy1].[X3] AS [RACE],
[GroupBy1].[X4] AS [C2]
FROM (SELECT [Extent1].[SEX] AS [X1],
[Extent1].[HISPANIC] AS [X2],
[Extent1].[RACE] AS [X3],
COUNT(1) AS [X4]
FROM [dbo].[DEMOGRAPHIC] AS [Extent1]
WHERE (EXISTS (SELECT 1 AS [C1]
FROM [dbo].[ENCOUNTER] AS [Extent2]
WHERE ([Extent1].[PATID] = [Extent2].[PATID]) AND ([Extent2].[ADMIT_DATE] >= @PrimaryObservationStart) AND ([Extent2].[ADMIT_DATE] <= @PrimaryObservationEnd))
AND (EXISTS (SELECT 1 AS [C1]
FROM [dbo].[DIAGNOSIS] AS [Extent3]
LEFT OUTER JOIN [dbo].[ENCOUNTER] AS [Extent4] ON [Extent3].[ENCOUNTERID] = [Extent4].[ENCOUNTERID]
WHERE ([Extent1].[PATID] = [Extent3].[PATID]) AND ([Extent4].[ADMIT_DATE] >= @PrimaryObservationStart) AND ([Extent4].[ADMIT_DATE] <= @PrimaryObservationEnd) AND ([Extent3].[CODETERM] = @CriteriaOneCodeType) AND ([Extent3].[CODETYPE] = @CriteriaOneMinimumAge))
) AS [Extent2]
```
Summary

- The MDQ tool includes a point and click, scalable query interface that supports complex logic for users to define cohorts of interest (e.g. medical codes, date ranges) for use in distributed queries.

- MDQs can be executed in various technical ecosystems.

- The tool is modularized, enabling functionality to be database-agnostic and can be run against multiple RDBMS platforms without custom programming as it utilizes widely adopted data exchange formats.
MDQ testing and validation process has been implemented and has shown consistent, valid results across database platforms; this process has informed tool development and continuous enhancements. As further real-world evidence for the tool, over 4,300 unique MDQ-to-site requests have been submitted within PCORnet, a DDN, since early 2016.

The tool is helping to close gaps by creating more opportunities for investigators to ask research questions more easily, flexibly, and rapidly within their DDNs while adhering to their local governance and technology policies.

Work continues to improve the MDQ tool in ways that will enable even more scalability in designing re-usable query interfaces and electronic workflows.
Acknowledgements

- PMN team at Harvard Pilgrim Health Care Institute
  - Jeffrey Brown
  - Chayim Herzig-Marx
  - Kimberly Barrett
  - Zachary Wyner
  - Adam Paczuski

- Software development partners
  - Daniel Dee
  - Dean Corriveau

- PCORnet Coordinating Center
  - Query Fulfilment Team
  - Data Characterization Team
  - Program Management Office