



Clinical Quality Language: 101
Wednesday April 13th, 2016
3:00 PM EDT

Deborah Krauss
Centers for Medicare & Medicaid Services

Bryn Rhodes
ESAC

Presentation Goals

- What is Clinical Quality Language (CQL)?
- Why was it developed?
- What problems does it solve and how?
- What can it do?
- What does it look like?
- Where can I get it?
- Where can I find more information?

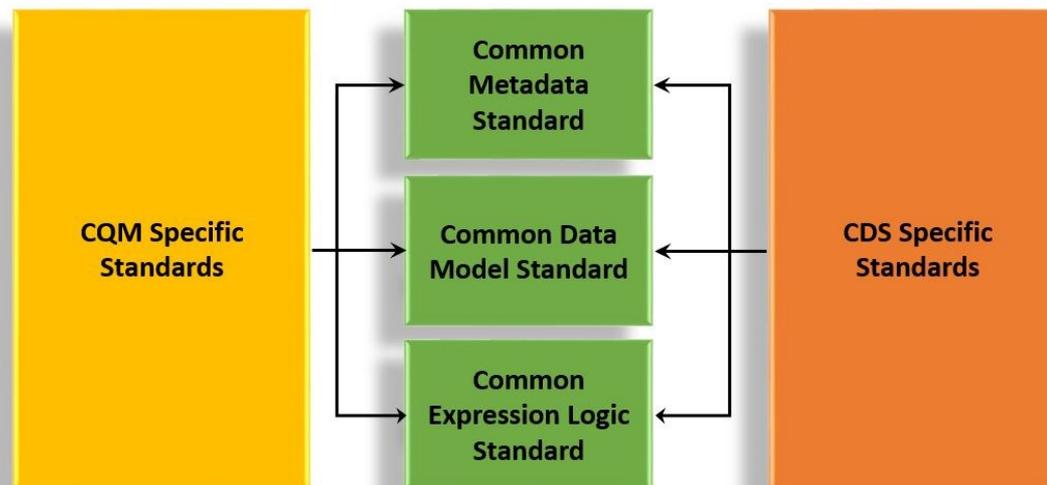
Sharing Clinical Knowledge

- Various means for representing Clinical Knowledge in an electronic format
 - Measurement
 - Quality Measures (Health Quality Measure Format -HQMF)
 - Guidelines (GEM, PDF)
 - Decision Support (CDS-KAS)
 - Event-Condition-Action (ECA) Rules
 - Documentation Templates
 - Order Sets
- How can we enable *computable* representations so we can automate as much as possible sharing and implementation of clinical knowledge?

Clinical Quality Framework (CQF)

Current specifications have different representations for the same concepts. Clinical Quality Framework has been working on aligning the specifications so that they use the same representations.

Decompose the problem of artifact representation into three components and build common specifications that can be used in both domains.



Clinical Quality Language (CQL)

- Health Level 7(HL7) standard designed to:
 - Enable automated point-to-point sharing of executable clinical knowledge
 - Provide a clinically focused, author-friendly, and human-readable language
- Currently a Draft Standard for Trial Use (DSTU) publication
 - http://www.hl7.org/implement/standards/product_brief.cfm?product_id=400

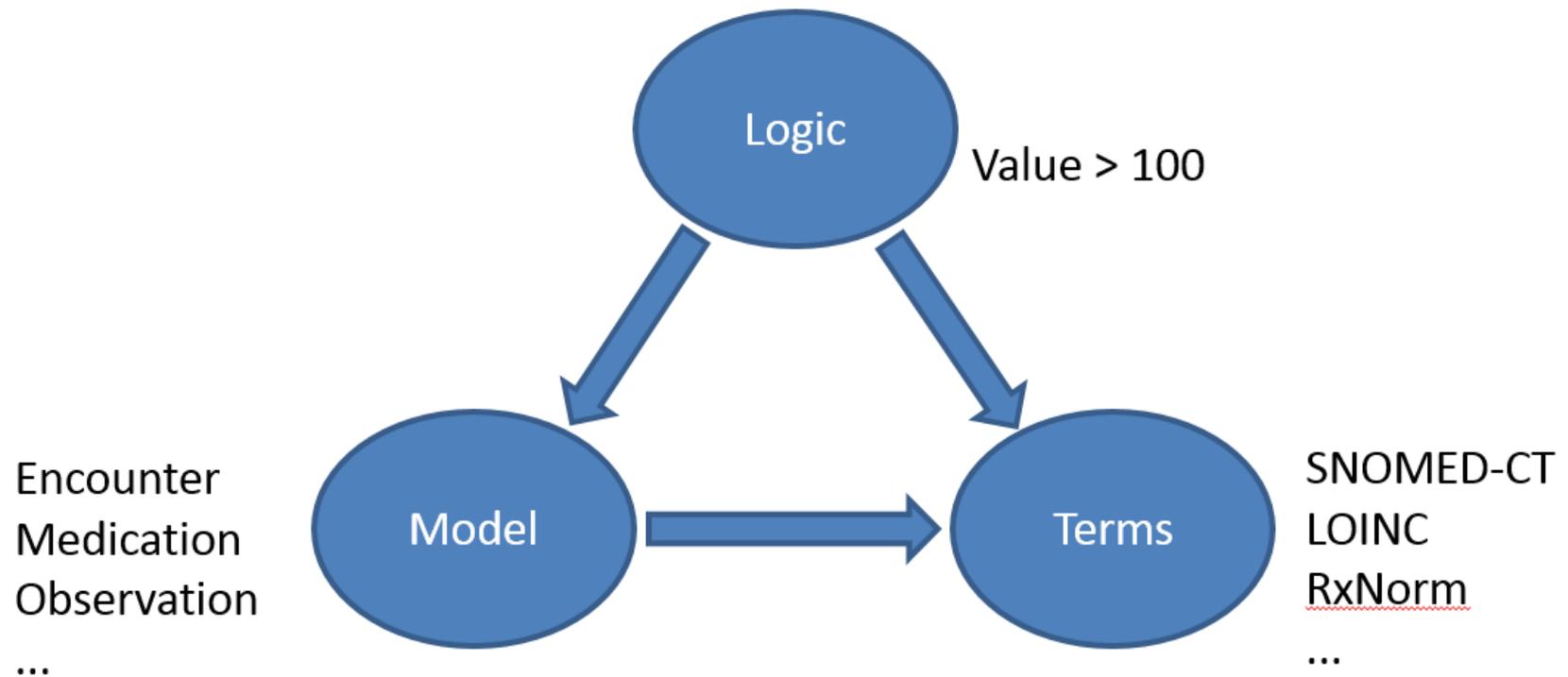
CQL Specification Target Audiences

- **Authors** – Clinical domain experts and clinical artifact authors
- **Developers** – Authors building more complex artifacts as well as shared libraries
- **Integrators** – Health-IT professionals integrating quality artifacts
- **Implementers** – Systems analysts, architects or developers building language processing applications

CQL Specification Content

- **Author's Guide** – Self-contained introduction to the language targeted at clinical quality authors
- **Developer's Guide** – More in-depth look at the language targeted at developers familiar with traditional development languages such as Java, C#, and SQL
- **Formal Specifications** – Logical and physical representation, as well as intended language semantics
- **CQL Reference** – A complete reference for all operators and functions in CQL

Components of Sharing Logic



Using CQL to Enable Sharing

- Quality Measurement Standards
 - HQMF – CQL-Based HQMF IG
 - Uses QDM as the model
 - Only replaces the *logic* representation of QDM, not the data structures
- Decision Support Standards
 - CDS Knowledge Artifact Specification (KAS)
 - Uses vMR as the model
 - Uses CQL to represent logic within the rules, order sets and documentation templates

CQL Library

- Named, versioned groupings of CQL components

```
45
46 library CMS55 version '1'
47
48 using QDM
49
50 valueset "Inpatient": '2.16.840.1.113883.3.666.5.307'
51
52 parameter "Measurement Period" default Interval[@2014-01-01T00:00:00.0, @2015-01-01T00:00:00.0)
53
54 context Patient
55
56 define "$Inpatient Encounters":
57     ["Encounter, Performed": "Inpatient"] E
58     where E.lengthOfStay <= 120 days
59     and E.dischargeDatetime during "Measurement Period"
60
```

Terminology in CQL

- Code systems and value sets used in CQL
 - Does not define the value set, only allows it to be referenced using an author-friendly name

```
45
46 library CMS55 version '1'
47
48 using QDM
49
50 valueset "Inpatient": '2.16.840.1.113883.3.666.5.307'
51
52 parameter "Measurement Period" default Interval[@2014-01-01T00:00:00.0, @2015-01-01T00:00:00.0)
53
54 context Patient
55
56 define "$Inpatient Encounters":
57     ["Encounter, Performed": "Inpatient"] E
58     where E.lengthOfStay <= 120 days
59     and E.dischargeDatetime during "Measurement Period"
60
```

CQL Parameters

- Named values that can be provided when the measure is evaluated
 - Measurement period for a quality measure
 - A1C threshold for an ECA rule

```
45
46 library CMS55 version '1'
47
48 using QDM
49
50 valueset "Inpatient": '2.16.840.1.113883.3.666.5.307'
51
52 parameter "Measurement Period" default Interval[@2014-01-01T00:00:00.0, @2015-01-01T00:00:00.0)
53
54 context Patient
55
56 define "$Inpatient Encounters":
57     ["Encounter, Performed": "Inpatient"] E
58     where E.lengthOfStay <= 120 days
59     and E.dischargeDatetime during "Measurement Period"
60
```

CQL Expressions

- Named expressions that define the logic of the artifact
 - Criteria definitions within a measure
 - Condition logic within an ECA rule
 - Behavior within an Order Set or Documentation Template

```
45
46 library CMS55 version '1'
47
48 using QDM
49
50 valueset "Inpatient": '2.16.840.1.113883.3.666.5.307'
51
52 parameter "Measurement Period" default Interval[@2014-01-01T00:00:00.0, @2015-01-01T00:00:00.0)
53
54 context Patient
55
56 define "$Inpatient Encounters":
57     ["Encounter, Performed": "Inpatient"] E
58     where E.lengthOfStay <= 120 days
59     and E.dischargeDatetime during "Measurement Period"
60
```

Context in a CQL Library

- Determines the level at which the artifact expressions operate
 - **Patient** – The expression is evaluated with respect to a single patient
 - **Population** – The expression is evaluated with respect to the entire population

```
45
46 library CMS55 version '1'
47
48 using QDM
49
50 valueset "Inpatient": '2.16.840.1.113883.3.666.5.307'
51
52 parameter "Measurement Period" default Interval[@2014-01-01T00:00:00.0, @2015-01-01T00:00:00.0)
53
54 context Patient
55
56 define "$Inpatient Encounters":
57     ["Encounter, Performed": "Inpatient"] E
58     where E.lengthOfStay <= 120 days
59     and E.dischargeDatetime during "Measurement Period"
60
```

Values in CQL

- Simple Types
 - Boolean, String, Number, Date/Time
 - true, 16, 'female', @2015-05-01
- Clinical Types
 - Quantities, Value Sets
 - 3 months, 6 'gm/cm3'
 - "Female Administrative Sex"
- Structured Types
 - Model Classes, Tuples
 - [Encounter]
 - Tuple { Name: 'Patrick', DOB: Date(2014, 1, 1) }
- List Types
 - { 1, 2, 3, 4, 5 }
- Interval Types
 - Interval(today - 1 years, today]

CQL Expressions

- Logic
 - A and B
 - A and not (B or C)
- Comparison
 - A >= B
 - A <> B
- Arithmetic
 - A + B
 - A + B * C

Timing and Intervals in CQL

- Full set from QDM
 - starts before start, starts same day as
- Timing phrases
 - starts 3 days before start
 - starts 3 days or less before start
 - starts within 3 days of start
- Interval operators
 - meets, overlaps, during
- Boundary access
 - start of MeasurementPeriod
- Membership
 - X in interval[4, 6]

Interval Operators in CQL

Operator/Inverse	Diagram	Interpretation
X same as Y Y same as X		start of X = start of Y and end of X = end of Y
X before Y Y after X		end of X < start of Y
X meets before Y Y meets after X X meets Y		successor of end of X = start of Y
X overlaps before Y Y overlaps after X X overlaps Y		start of X <= start of Y and start of Y <= end of X
X begins Y		start of X = start of Y and end of X <= end of Y
X included in (during) Y Y includes X		start of X >= start of Y and end of X <= end of Y
X ends Y		start of X >= start of Y and end of X = end of Y

Date/Time Arithmetic in CQL

- Date construction
 - `@2014-01-01T12:00:00-06:00`
 - `Date(2014, 1, 1, 12, 0, 0, -6)`
 - `convert '2014-01-01T12:00:00-06:00' to DateTime`
- Date arithmetic
 - `Today() + 3 months - 2 days`
 - `months between start of X and end of X`
 - `difference in days between X and Y`
 - `duration in months of X`
- Date/Time extraction
 - `date from D // returns the date without the time`
 - `time from D // returns the time without the date`
- Component extraction
 - `month from D // returns the number of whole units`

CQL Retrieve

- All data access in CQL is performed with the *retrieve*
- Simplest case retrieves all records

```
55  
56 define "$Inpatient Encounters":  
57     ["Encounter, Performed"]  
58
```

- Typically restricted to a value set
 - Primary attribute is defined by the model

```
55  
56 define "$Inpatient Encounters":  
57     ["Encounter, Performed": "Inpatient"]  
58
```

- Can explicitly specify the attribute as well

```
55  
56 define "$Inpatient Encounters":  
57     ["Encounter, Performed": code in "Inpatient"]  
58
```

CQL Queries

- The *query* construct is used to perform various operations, including filtering, shaping, sorting, and relating results
- Simplest query involves only a single source

```
55  
56 define "$Inpatient Encounters":  
57     ["Encounter, Performed": "Inpatient"] E  
58
```

- The alias “E” allows the source to be referenced anywhere within the query

Filtering in CQL Queries

- A *where* clause returns only those elements that satisfy the condition

```
55  
56 define "$Inpatient Encounters":  
57     ["Encounter, Performed": "Inpatient"] E  
58     where E.lengthOfStay <= 120 days  
59
```

- Can include multiple conditions

```
55  
56 define "$Inpatient Encounters":  
57     ["Encounter, Performed": "Inpatient"] E  
58     where E.lengthOfStay <= 120 days  
59         and E.dischargeDatetime during "Measurement Period"  
60
```

Relationships in CQL Queries

- Queries can include *with* and *without* to define relationships to other data

```
62  
63 define "$Emergency Department Encounters":  
64     ["Encounter, Performed": "Emergency Department Visit"] ED  
65     with "$Inpatient Encounters" E  
66     such that ED.dischargeDatetime 1 hour or less before E.admissionDatetime  
67
```

Shaping in CQL Queries

- The *return* clause can be used to calculate results

```
69  
70 define "$Measure Observation":  
71     "$Emergency Department Encounters" E  
72     .....  
73     where E.admissionDatetime is not null  
74           and E.dischargeDatetime is not null  
75     return minutes between E.facilityLocationArrivalDatetime and E.facilityLocationDepartureDatetime
```

Population Context in CQL

- The *Population* context indicates the expression will be evaluated at the population, rather than the individual level

```
72  
73 context Population  
74  
75 define "$Measure Score": Median("$Measure Observation")  
76
```

- CQL has functions for all the standard statistical aggregates

Measure Library for CMS 55v1

```

45
46 library CMS55 version '1'
47
48 using QDM
49
50 valueset "Inpatient": '2.16.840.1.113883.3.666.5.307'
51 valueset "Emergency Department Visit": '2.16.840.1.113883.3.117.1.7.1.293'
52
53 parameter "Measurement Period" default Interval[@2014-01-01T00:00:00.0, @2015-01-01T00:00:00.0)
54
55 context Patient
56
57 define "$Inpatient Encounters":
58     ["Encounter, Performed": "Inpatient"] E
59     where E.lengthOfStay <= 120 days
60     and E.dischargeDatetime during "Measurement Period"
61
62 define "$Emergency Department Encounters":
63     ["Encounter, Performed": "Emergency Department Visit"] ED
64     with "$Inpatient Encounters" E
65     such that ED.dischargeDatetime 1 hour or less before E.admissionDatetime
66
67 define "$Measure Observation":
68     "$Emergency Department Encounters" E
69     where E.admissionDatetime is not null
70     and E.dischargeDatetime is not null
71     return minutes between E.facilityLocationArrivalDatetime and E.facilityLocationDepartureDatetime
72
73 context Population
74
75 define "$Measure Score": Median("$Measure Observation")
76

```

CQL Resources

- HL7 Standard: Clinical Quality Language Specification, Release 1 DSTU
 - http://www.hl7.org/implement/standards/product_brief.cfm?product_id=400
- HL7 CDS Workgroup Project Homepage:
 - http://wiki.hl7.org/index.php?title=Clinical_Quality_Language
- GitHub Tools Repository:
 - https://github.com/cqframework/clinical_quality_language

What's Next

- Upcoming CQL Events
 - Cooking with CQL Teaser: April 21st
 - CQL Training for Measure Developers: April 27th
 - Cooking with CQL: or How to Incorporate CQL into HQMF for eCQMs: April 28th
- Find more information on the eCQI Resource Center CQL webpage
 - <https://ecqi.healthit.gov/cql>

Questions?