CQL & BPM+

Considerations and approaches
Deliverable for TAC-19-56435

Prepared for the September 2019 OMG Technical Meeting by Bryn Rhodes, CTO, Dynamic Content Group

# Overview

This white paper is an accompanying briefing for the presentation of the same title, delivered to the BPM+ Health Workgroup at the OMG Technical Meeting in Nashville, TN on September 23rd, 2019. The paper provides a brief introduction to the Health Level 7 (HL7) International Clinical Quality Language (CQL) specification, where it is used, as well as considerations and approaches on how it might be used within the BPM+ Health standards ecosystem. Finally, the paper recommends potential next steps for continued exploration and engagement.

# What is CQL?

As background for this paper and the accompanying presentation, we begin with a brief, non-controversial statement of purpose:

***To reduce the development effort required for clinical systems to support evolving standards of care***

We believe this is a major shared goal of the standards development process across BPM+ Health, HL7, and other health-related standards development organizations. However, several points should be made about this goal: it requires that we reduce duplicate effort while still supporting innovation; it requires that we reduce inconsistency without ignoring settings-specific factors; and it demands that we reduce *accidental complexity*, so that we can focus on the *essential complexity* inherent in the application of technology to clinical practice. Those technologies currently take many forms, including guidelines, interventions, quality measures, documentation templates, order sets, flow sheets, SMART-on-FHIR applications, CDS Hooks services, to name a few.

To help tame some of this complexity, model-driven architectures and approaches have evolved around several key principles: that a *declarative* approach is generally better than an *imperative* one (i.e. “what”, not “how”); that *separation of concerns* is a critical principle of software and system design; that various types of *independence* are of paramount importance to ensuring modularity and longevity; and that the history of computing is nothing if not a history of *raising the level of abstraction*. At the same time, these approaches must also recognize that there is no silver bullet. Software and system design is an inherently challenging problem, all the more so in the healthcare space where so much is at stake.

This goal and these principles have been at the heart of the Health eDecisions initiative, and its successor, the Clinical Quality Framework initiative, now a joint project of the Clinical Decision Support (CDS) and Clinical Quality Improvement (CQI) HL7 Work Groups. Following these principles leads quite naturally to the following, we hope equally non-controversial goal:

***To enable precise and unambiguous sharing of clinical logic***

This sub-goal is one of the primary design requirements of the Clinical Quality Language specification. Of course, the specification was not developed in a vaccuum, nor was it developed without due consideration for existing expression language formalisms. There were (and still are) many, but each of the available standards failed to meet fully the design requirements of being at once:

1. Clinician and domain-expert friendly
2. Clinically focused
3. Unambiguous
4. Platform and technology independent
5. Sufficiently expressive

As such, the Clinical Quality Framework initiative worked with stake-holders from across the industry to produce a formalism that could enable automated, point-to-point sharing of executable clinical knowledge, while at the same time providing a clinically focused, author-friendly, and human-readable language.

Briefly, CQL is a *query* language, meaning that it supports at its core a declarative query construct, similar to other data query languages such as Structured Query Language (SQL). The syntax is deliberately similar enough to SQL to be familiar, but intentionally calls out specific operations that are ubiquitous in healthcare logic.

CQL is also *pure functional*, meaning that it has no side-effects. It is not possible to change data with CQL; expressions and queries of CQL can only be evaluated and return results. Changing data must be done with mechanisms outside of CQL.

And finally, CQL is intentionally a *conceptual* level language, meaning that it provides a complete separation from any particular platform or technology, and many of the design choices made in the language are explicitly about facilitating implementation across platforms.

# Where is CQL Used?

CQL is currently published as a Standard for Trial Use (STU) by Health Level 7 (HL7) International. The first STU publication was in May 2015, and since that time the specification has undergone three subsequent ballots, incorporating feedback from implementation experience across a broad range of use cases. The resulting current STU 4 publication is a mature and capable specification that has been adopted by the Centers for Medicare and Medicaid (CMS) for use in specifying electronic Clinical Quality Measures (eCQMs) as of the 2018 annual update. In addition, CQL has been piloted for use in a number of exploratory projects, including pharmacogenomics, all-hazards response, Centers for Disease Control (CDC) opioid prescribing support, radiology appropriateness, immunization decision support, World Health Organization (WHO) antenatal care guidelines, and many others.

In addition to a healthy open source community with multiple active evaluation and tooling projects, at least one vendor has a native CQL evaluation product that is being used for quality reporting in hospitals across the United States, and we are aware of several others that have products in pilot or near-release state.

# Why use CQL?

The lineage and design focus of CQL ensures that it is fit-for-purpose within the clinical domain, not only for quality measurement and decision support, but for defining research cohorts, phenotyping characteristics, public health reporting, and others. This broad base of use cases enables clinical logic written in CQL to be re-used, as well as shared. For example, several quality measures are currently re-using Morphine Milligram Equivalent calculation logic from complementary clinical decision support rules.

In addition, as the maturity and adoption of CQL grows, so too does the community of users around it, including quality measure and decision support authors, vendors, and toolsmiths, providing an increasing pool of knolwedge engineering and technical expertise with CQL.

# CQL & BPM+

The BPM+ Health ecosystem is currently comprised of three specifications: BPMN, used for deterministic and structured workflows; CMMN, used for non-deterministic or unstructured workflows; and DMN, used for modeling decisions within those workflows.

Both BPMN and CMMN support the ability to describe expressions at several key points throughout the specifications. Both specifications model this capability in a similar way, with a structure that allows any string of any language to be used. Because the CQL specification defines a media type, *text/cql*, this capability can be used to easily introduce CQL expressions within any BPMN or CMMN workflow.

Within BPMN, the *FormalExpression* construct can be used for this purpose. In CMMN, the *Expression* construct supports this.

However, although the DMN specification supports this same capability with the *LiteralExpression* construct, it also explicitly requires the use of S-FEEL (for Conformance Level 2) and FEEL (for Conformance Level 3). It’s not clear why this requirement exists, and it may well be that the next version of DMN has already addressed this issue, but as it is, DMN would not support the use of CQL (or any other expression language). Apart from that issue, it seems that the DMN specification could support the use of any expression language, in the same way that BPMN and CMMN already do.

However, a significant challenge to the use of CQL in this way is that CQL is model-agnostic, relying on model definitions (called *model info*)provided as input to the CQL tooling to validate that the CQL is correct for the model. With BPMN, CMMN, and DMN, the data requirements for each artifact are defined as part of the artifact itself. This would effectively mean that a distinct model info would need to be created for each artifact. This presents a technical as well as logical barrier to both sharing the artifact as well as reuse of clinical logic between artifacts. Even in the best case of a set of artifacts using consistent model definitions, these definitions would be duplicated in every artifact in which they appeared.

# CQL & DMN

The DMN specification defines an expression language called Friendly Enough Expression Language (FEEL), as well as a simplified subset of that expression language (called Simplified FEEL, or S-FEEL). As an expression language, CQL and FEEL have significant overlap in functionality. Over the past year, the editors of CQL and DMN, together with a few members of the BPM+ community, have been reviewing the specifications to determine the boundaries of that overlap, with the goal of identifying and closing gaps and looking for opportunities for alignment.

Although incomplete, the initial results of that investigation are that the two languages are remarkably similar semantically, and that with very few exceptions, it should be possible to convert between the two languages. There are some caveats to doing so (for example, FEEL considers the *null* marker to be a member of the domain of every type, whereas CQL considers the *null* marker to be outside of any type, resulting in some subtle semantic differences), but it appears that this translation could be effective, albeit with the need to define some additional functions in one or the other language.

This semantic alignment presents potential BPM+ adopters with two possibilities:

1. Use DMN with CQL directly, enabling reuse and sharing of CQL expressions throughout and across BPM+ artifacts.
2. Translate between FEEL and CQL, enabling the use of CQL artifacts in FEEL-enabled environments and vice-versa.

In other words, there is no technical reason that CQL and FEEL could not be used interchangeably within the DMN specification.

# What Next?

Potential go-forward steps include:

## Exploratory Group

We recommend the formation of a small group of focused and relevant experts to explore how the recommendations in this paper might be applied to the BPM+ Health specifications. This group could begin with the goal of representing an exemplar guideline with BPM+ Health specifications and Clinical Quality Language. Other than perceived violation of a narratively enforced conformance requirement in DMN, this project could proceed without necessarily modifying any of the underlying specifications, and the exercise would, we hope, inform future direction of the specifications.

## Engage with Specific Projects

We recommend that the BPM+ Health effort continue to support engagement with specific projects such as the CDC’s CDS Kaizen initiative. This engagement has already been productive from the CDS Kaizen perspective, and we hope that it has likewise been productive for the BPM+ Health community.

## Support Cross-Pollination (OMG/HL7)

We recommend that the OMG and HL7 continue supporting cross-pollination efforts between the two organizations. In particular, the BPM+ Health community and HL7 Work Groups would mutually benefit from additional cross engagement.

# Other Considerations

As part of the analysis performed for this project, it became clear that BPMN, CMMN, and DMN could all benefit significantly by closing two important gaps:

1. Terminology – The specifications should be enhanced to support the use of concepts throughout
2. Data Model – The specifications should be enhanced to support the ability to define and reference a common data model throughout

We recommend strongly that the HL7 Fast Healthcare Interoperability Resources (FHIR) specification be referenced to meet both these requirements.