Digital Professionalism in Health and Care: Developing the Workforce, Building the Future
P. Scott et al. (Eds.)

doi:10.3233/SHTI220914

Experiences of Creating Computable Knowledge Tutorials Using HL7 Clinical Quality Language

Elisavet ANDRIKOPOULOU[a,1] and Philip SCOTT[b]
[a] *University of Portsmouth*
[b] *University of Wales Trinity Saint David*

Abstract. Computable knowledge artefact development is challenging and often culminates in the development of unique single usage solutions. Libraries of computable knowledge artefacts have the possibility to enhance the Learning Health System in order to improve the benefits of innovation and the decision making of clinicians. This paper aims to discuss the process of creating the use cases and the tutorial material that would enable students to both understand how the interaction between the dataset and the outcome occurs as well as how HL7 Clinical Quality Language can be used to create artefacts of re-usable code.

Keywords. Clinical Quality Language, tutorial, computable knowledge, artefact

1. Introduction

The National Institute for Health and Care Excellence (NICE) is responsible for developing healthcare guidelines for England and Wales based on current knowledge. This type of knowledge typically derives from narrative research articles, which may be included in a systematic literature review after 5-15 years of publication [1]. Using these guidelines developers are able to construct computable knowledge artefacts, to enhance clinical decision support programs, such as early warning scores and medical device alerts [1].

Computable knowledge artefact development is challenging as there is no “golden rule” solution and a number of developed systems do not meet NICE guidelines and needs [2]. Computable knowledge is an important element in a Learning Health System (LHS), since knowledge that is produced using healthcare data can be transformed into machine-executable arguments and represented using standardized formats and mathematical approaches [3]. Health Data Research (HDR) UK also recognizes the importance of LHS and funded the Better Care programme [4], focusing on person-centered care for better outcomes, which supports the data informed healthcare decision making. The HDR UK Better Care programme ran a “collaborathon” (a form of ‘hack day’) to explore how their research knowledge could be represented in computable forms. However, few computable knowledge artefacts were identified in the Better Care

[1] Corresponding Author, Corresponding author, School of Computing, University of Portsmouth, UK; E-mail: elisavet.andrikopoulou@port.ac.uk.

initiative and although there are open source-code repositories such as GitHub, these focus on data processing and not knowledge generation [5].

Sharing computable knowledge is the fundamental next step following the creation of open source code that although functional are stand alone and not re-used actionable insights that could enable scaling [5]. There is limited mentioning of how an app developer or people related to LHS can develop such open source re-usable code fragments, often known as libraries.

This paper aims to discuss the process of creating the use cases and the tutorial material that would enable the students to both understand how the interaction between the dataset and the outcome occurs as well as how HL7 Clinical Quality Language (CQL) [6] can be used to create artefacts of re-usable code. These artefacts were developed using the HDR UK collaborathon as a starting point. Future deployment is planned in a collaborathon series for NICE.

2. Methods

The first step was to create the use cases that provide the data and define the requirements. Two different long-term conditions were selected, due to familiarity of the authors with them. The first use case focused on type 2 diabetes (T2D) diagnosis based on the Type 2 Diabetes diagnostic NICE guidelines [7]. The second use case focused on prescribing direct oral anticoagulants (DOAC) in patients with Non-Valvular Atrial Fibrillation (AF) and for the treatment and prevention of venous thromboembolism (VTE) [8].

The functional requirements [9] for the CQL software tutorials are shown in Table 1. These requirements implemented within these use cases focus on the structure, semantics, and encoding of expression logic representation within the CQL artefacts. Two different experienced general practitioners face verified these use cases to ensure applicability in the real life scenarios.

Table 1. CQL software functional requirements

Input parameters	Behaviors	Outputs
Patient demographics	Is value in range?	Diagnosis of T2D
Blood tests	Are there comorbid conditions?	Prescribing of DOAC
Patient history	Is there history risk?	Alter the prescription of DOAC
Current treatment	Is medication correct?	
Other medication	Can medication change?	

Based on the use cases and the functional requirements we identified the key data concepts that we needed to store to use as bases of testing our artefacts. One of the most important and difficult aspects of testing whether an artefact of computable clinical knowledge is usable it to find data that are both purposeful, anonymized and comprehensive enough. Creating a synthetic dataset of patient data is not easy nor likely to appear truthful. The data need to be plausible and convincing as well as mapping truly to the use case each artefact is working on. There is a need of a control dataset that would also include plausible and convincing data but will not be truly applicable on the use case.

We used the International Classification of Diseases (ICD-10-CM) [10] and SNOMED CT [11] code systems to define the codes that correspond to diseases and patient symptoms. These codes were also validated by the GPs, for applicability and to enhance plausibility.

Finally, to host the data and queries in an academic friendly and replicable environment we selected to build a FHIR (Fast Healthcare Interoperability Resources) [12] server in Amazon AWS. FHIR servers are globally popular, well documented and support both data and knowledge artefacts. To query the server, we selected the CQL (Clinical Quality Language) [6] as it is being co-developed with FHIR. CQL is an authoring programming language similar to the much more common web authoring languages such as the structural HTML and the scripting Javascript and PHP [6, 13].

3. Results

The produced CQL artefacts based on these use cases followed the coding structure and syntax diagram, which is illustrated in Figure 1 and is based on the CQL guidelines.

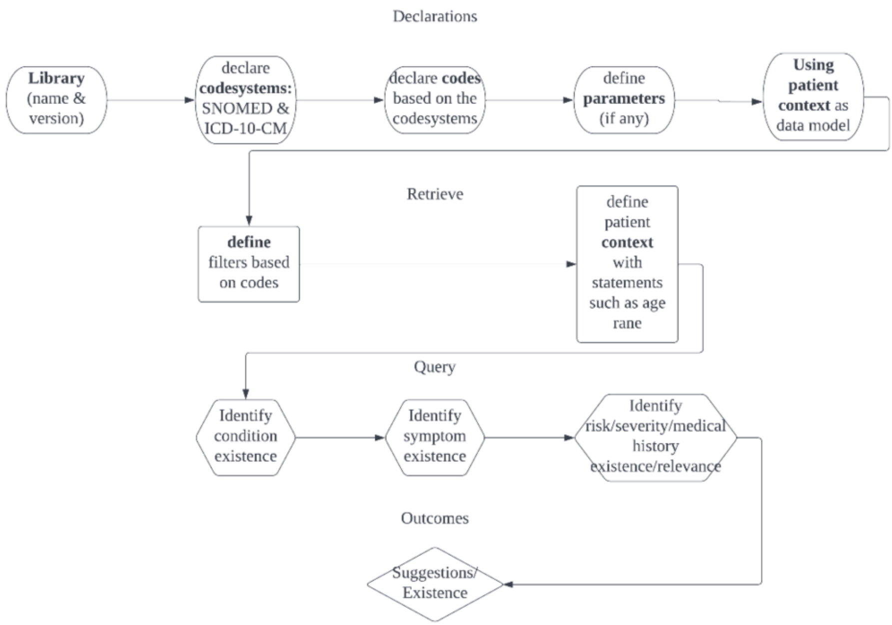

Figure 1. CQL syntax diagram.

To create the CQL artefacts from the use cases we had to implement the syntax as shown in Figure 1 and to develop the algorithm, identify the key data items needed to execute the algorithm and identify the particular SNOMED CT values to trigger the decision points. For example, in Figure 2 there is an initial simplified CQL developed fragment that was used as a first step in the use case implementation:

```
code "Condition Active code": 'active' from "CONDCLINSTATUS" display 'Active'
code "Condition Confirmed code": 'confirmed' from "CONDVERSTATUS" display 'Confirmed'

code "Dizziness (SNOMED)": '404640003'  from "SNOMED" display 'Dizziness SNOMED'
code "Gastrointestinal symptoms (SNOMED)": '267045008'  from "SNOMED" display
'Gastrointestinal symptoms SNOMED'

concept "Condition Active": { "Condition Active code" } display 'Active'
concept "Condition Confirmed": { "Condition Confirmed code" } display 'Confirmed'

context Patient

define "AgeRange-old at end of Measurement Period":
    AgeInYearsAt(end of "Measurement Period") between 60 and 110

define "AgeRange-middle at end of Measurement Period":
    AgeInYearsAt(end of "Measurement Period") between 20 and 59

define "AgeRange-young at end of Measurement Period":
    AgeInYearsAt(end of "Measurement Period") < 19

//test if patient has the symptoms
define "HasDizziness":
    exists(ActiveCondition([Condition: "Dizziness (SNOMED)"]))
define "HasGastrointestinalSymptoms":
    exists(ActiveCondition([Condition: "Gastrointestinal symptoms (SNOMED)"]))

//Shall we suggest further testing?
  define "PotentialToHave":
    "HasDizziness" or "HasGastrointestinalSymptoms"
```

Figure 2. CQL code fragment

Translated, the first two lines define a condition as active and diagnosed. The next two lines create the labels “Dizziness (SNOMED)” and “Gastrointestinal symptoms (SNOMED)” to link the concepts with the SNOMED CT IDs. The next two lines are used as shorthand for the active and confirmed conditions. The next line defines the context as patient, followed by the definition of age range as old, middle and young. The last two lines test whether a patient “HasDizziness” and “HasGastrointestinalSymptoms”, which are triggered if the patient record has the appropriate SNOMED code for those, and a suggestion is made of the need for further testing if the patient has these symptoms.

4. Discussion

The creation of the use cases and the CQL artefacts for the tutorials were challenging and many lessons were learned. The actual computable knowledge artefacts and their final testing will be published in the future, as they are still being developed. This project aimed to discuss the process of creating the use cases and to create the tutorial material that would enable the students to both understand how the interaction between the dataset and the outcome occurs as well as how CQL can be used to create artefacts.

As part of an active learning approach [14] that enables students to be more involved and engaged in their learning we developed realistic and plausible use cases to enhance student’s experimentation and problem-solving capabilities in a convincing dataset.

Besides focusing on the active learning approach, learner characteristics also can influence their education achievements, to ensure knowledge acquisition, students’ skills and experiences should be inclusive and accessible [15]. The developed tutorials target graduates that have a health informatics background and therefore are either able to code or are healthcare policy makers or medically trained professionals. The CQL artefacts follow a step-by-step approach as is illustrated in Figure 1 to enable students without strong coding skills to understand and follow the programming logic. The next iteration of these tutorials will be used to support the forthcoming series of collaborathons organized by NICE.

References

[1] Scott P (2021) Computable knowledge Part 1 of 2: What is it and why do we need it to build a Learning Health System? In: HDR UK. https://www.hdruk.ac.uk/news/computable-knowledge-part-1-of-2-what-is-it-and-why-do-we-need-it-to-build-a-learning-health-system/. Accessed 30 Jun 2022

[2] Mitchell A (2020) A NICE perspective on computable biomedical knowledge. BMJ Heal Care Informatics 27:e100126. https://doi.org/10.1136/bmjhci-2019-100126

[3] MCBK Community (2021) What is computable biomedical knowledge? https://mobilizecbk.med.umich.edu/. Accessed 30 Jun 2022

[4] NHS England (2017) Better Care Fund

[5] Scott P (2021) Computable knowledge Part 2 of 2: learning from the HDR UK collaborathon – make it FAIR!

[6] HL7 International Clinical Quality Language (CQL). https://cql.hl7.org/. Accessed 30 Jun 2022

[7] NICE (2022) When should I suspect type 2 diabetes in an adult? In: Clin. Knowl. Summ. https://cks.nice.org.uk/topics/diabetes-type-2/diagnosis/diagnosis-in-adults/. Accessed 30 Jun 2022

[8] NHS Greater Glasgow and Clyde Medicines Information Service (2018) DOAC Prescribing in Patients with Non-Valvular AF and for the treatment and prevention of VTE Frequently Asked Questions

[9] Bigelow SJ (2020) Functional vs. nonfunctional requirements in software engineering. https://www.techtarget.com/searchsoftwarequality/answer/Functional-and-nonfunctional-requirements. Accessed 30 Jun 2022

[10] Centers for Disease Control and Prevention (CDC) (2015) International Classification of Diseases, (ICD-10-CM/PCS) Transition - Background. https://www.cdc.gov/nchs/icd/icd10cm_pcs_background.htm. Accessed 30 Jun 2022

[11] NHS Digital (2022) SNOMED CT. In: NHS Digit. https://digital.nhs.uk/services/terminology-and-classifications/snomed-ct. Accessed 30 Jun 2022

[12] HL7.org FHIR Specification. https://hl7.org/fhir/. Accessed 30 Jun 2022

[13] Assignmenthelp (2022) Authoring Languages. https://www.assignmenthelp.net/pg/authoring-languages. Accessed 30 Jun 2022

[14] UCL Arena Centre for Research-based Education (2022) Active learning. https://www.ucl.ac.uk/teaching-learning/publications/2019/aug/active-learning. Accessed 30 Jun 2022

[15] Garrison DR, Kanuka H (2004) Blended learning: Uncovering its transformative potential in higher education. Internet High Educ 7:95–105. https://doi.org/10.1016/j.iheduc.2004.02.001