AHRQ National Web Conference on the Clinical Decision Support Authoring Tool
Q&A Document

Q: When will we have resources of FHIR available. This was discussed at the last ONC conference, but we were told DaVinci info would be available. Thanks for presenting this information today.

The CDS Authoring Tool currently supports a subset of FHIR DSTU2 and STU3 resources, including AllergyIntolerance, Condition, Encounter, MedicationOrder (DSTU2 only), MedicationRequest (STU3 only), MedicationStatement, Observation, and Patient (age/gender), and Procedure. Other resources may be added in the future based on feedback from our users. As for DaVinci, we ask that you reach out to us via the "Feedback" link in the CDS Authoring Tool to clarify the question.

Q: What about system users - for automation? The automation would be for downloading the artifacts. No account needed.

In the CDS Connect Repository, the logic file attachments can be downloaded by anyone; no account is needed. The CDS Connect Repository also has an API, allowing artifact metadata to be retrieved programmatically. For more details on the CDS Connect Repository API, please contact the CDS Connect Team via the "Contact Us" link on the CDS Connect Repository.

The CDS Authoring Tool only allows artifacts to be exported as CQL and downloaded by the logged in author. The CDS Authoring Tool does not provide a capability to share artifacts within the tool, so it does not make sense for an automated system user to interact with the CDS Authoring Tool.

Q: What level of mathematical complexity can the Authoring Tool support (i.e. long equations for prediction models)?

The CDS Authoring Tool supports a subset of the CQL specification and is currently focused on CQL features that support establishing record-based selection criteria for determining if a patient should be provided a recommendation. The CDS Authoring Tool does not yet support any mathematical operators aside from comparisons of lab values against a known value. Future versions may support additional capabilities, particularly if there is a way to provide them in a user-friendly manner.

The CDS Authoring Tool team is also considering approaches toward allowing external CQL to be integrated into the CDS Authoring Tool. Such a feature might allow complex mathematical algorithms to be authored in an external environment, uploaded to the CDS Authoring Tool, and referenced by name.

Using the current capabilities, users might also choose to author as much as they can in the CDS Authoring Tool, export it to CQL, and complete the CQL development manually, adding in those capabilities that the CDS Authoring Tool does not support.

Q: In the age range element example, how to specify inclusive vs exclusive boundaries?

The Age Range component in the CDS Authoring Tool always uses inclusive boundaries. This allows the underlying CQL to use a consistent approach across artifacts.

Q: Who identifies the evidence to be used for the Artifacts? What is their process of appraisal to determine the quality of the evidence? Who are these people?

On the CDS Connect project, a group of physicians and RNs perform an environmental scan of evidence-based research across the clinical topic/domain that is targeted for CDS development, often pulling in subject matter experts (SMEs) and specialists in the clinical domain. Other organizations would likely include the same types of individuals (clinical

SMEs, CMO). Peer-reviewed evidence is preferable, as is evidence that uses the Grades of Recommendation Assessment, Development and Evaluation (GRADE) system to rate clinical practice guidelines. CDS development is usually focused on expressing GRADE A and B recommendations.

Q: What is the name of that capability? That provided the drop-down box of units.

When specifying comparisons for quantities, the CDS Authoring Tool presents a unit selector that assists the user in choosing a valid UCUM unit (as required by the CQL specification). This capability leverages the API for UCUM and an autocompleter, both provided by the Lister Hill National Center for Biomedical Communications at the U.S. National Library of Medicine.

Q: Very interesting application! Do you have a sense how many of the major EHR vendors will have APIs that allow providers to import these artifacts into their production environment?

EHR vendors are in various stages of progress for supporting CQL-based artifacts natively. Early CQL implementations will likely be focused on electronic Clinical Quality Measures (eCQMs) to support the CQL-based eCQMs for CMS's 2019 reporting period. Since these eCQMs use the Quality Data Model, however, EHRs will need to add support for FHIR-based CQL in order to be used effectively for CDS. If you are a customer of an EHR, we recommend you reach out to your EHR representative to ask about progress toward native support for CQL.

Until EHRs have native support for FHIR-based CQL logic, there are other approaches toward integrating CQL with an HER:

- Since many EHRs support SMART on FHIR, CQL logic could be embedded into a SMART on FHIR application and integrated into an EHR. CDS Connect's Pain Management Summary is an example of such an approach.

- As EHRs add support for CDS Hooks, CQL-based logic could be executed as part of a CDS Hooks service. The CQL Services prototype that was demonstrated at the end of the webinar is an example of this approach.
- Custom CDS integrations, using proprietary APIs, could leverage CQL execution engines or CQL execution services for evaluating the CQL logic. This approach is detailed in the 2017 CDS Connect Pilot Report as well as CDC's Opioid Prescribing Support Implementation Guideline.
- 3rd party CDS-focused solutions and platforms may also provide CQL-based capabilities.

Q: What is the CQL now being viewed in? thanks

During the demonstration, we viewed local CQL files using Visual Studio Code and a CQL extension. This CQL extension is available on GitHub, but is not published to the Visual Studio Marketplace, so it must be installed manually. There is another extension in the marketplace called "Clinical Quality Language", but neither MITRE nor AHRQ have installed or verified it. Use at your own risk.

There is also a CQL package for the popular Atom text editor. This CQL package has been used successfully by MITRE in the past and provides the same basic capabilities as the CQL text editor used in the demonstration.

Q: Plans to move beyond FHIR DSTU2?

The CDS Authoring Tool was initially developed to support FHIR DSTU2, as that was the version of FHIR most prevalent in the marketplace. Support for FHIR STU3 was recently added in response to requests from CDS Authoring Tool users. Authoring is the same for both versions, but now users can choose DSTU2 or STU3 when they download the CQL representation of the artifact. The CDS Authoring Tool team will continue to monitor industry adoption of FHIR and will add support for FHIR R4 if and when it makes sense.

Q: Is there a list of the steps to make sure none is missed

The CDS Authoring Tool provides a user guide with detailed explanations of each component in the CDS Authoring Tool. In addition, the CDS Authoring Tool presents dynamic warnings to authors to assist them in creating logically correct expressions.

Q: Can Synthea patients be made available for testing to the authors?

The CDS Authoring Tool currently requires that authors upload their own synthetic test patients. As noted in the demonstration, Synthea™ can be used to generate synthetic patients for this purpose. Synthea™ is a java-based application and requires local installation and configuration to run it. See the Synthea™ wiki for instructions.

If you're not able to run Synthea locally, you can download pre-built data sets of Synthea™ patients on the SyntheticMass website (choose the DSTU2 or STU3 formats; the FHIR 1.8.0 format is not supported).

In either case, you may need to search through the patients and/or modify the patient records to find patients suitable for the exact needs of testing your CDS logic.

Q: Is it possible to load externally authored cql artifacts for testing in the authoring environment?

Authors cannot currently upload externally authored CQL to the CDS Authoring Tool. The CDS Authoring Tool team is considering adding a similar feature, but the main goal of that feature would be to allow tool-authored CQL to reference externally-authored CQL (in order to provide features not yet supported by the CDS Authoring Tool).

The CDS Connect project team, however, is currently developing a stand-alone CQL testing framework to support building and executing CQL tests in other environments. We anticipate that a production level tool will be available by September.

Q: How are actions other than reminders to clinicians handled? (e.g., post orders, record observations, etc.)

The CDS Authoring Tool does not directly support specifying a CDS intervention/action "type" (e.g., reminder, order set, documentation template). Currently, the CDS Authoring Tool is focused on rules-based logic that returns recommendations. That said, the Base Elements tab could be used to author arbitrary expressions that may be useful in the context of other CDS types; but it's likely that additional modification to the CQL would be needed after download.

Regardless of the tool used to create the CDS, the CDS intervention/action "type" is generally selected based upon the recommendation that is being expressed as CDS, where/how it best integrates with clinical workflow, and the targeted end user (e.g., physician, RN, patient). The logic that drives the presentation of the intervention (e.g., the list of specific orders in an order set or the list of patient-level data in a dashboard intervention) is informed by the clinical research that occurs prior to starting CDS development. The required clinical concepts for the intervention/action are expressed using the FHIR data model and the logic is expressed using CQL.

Integrating the CDS expression (whether it be an alert, order set, documentation template, etc.) requires additional standards and capabilities (outside of CQL). The FHIR Clinical Reasoning standard, or in some cases CDS Hooks, can be used to formally specify the CDS intervention/action type and its details. This, of course, requires the EHR to support these specifications. Depending on the EHR, a custom integration may be necessary to provide seamless support for certain CDS action/intervention types. As the market matures, and standards are adopted, we hope that the need for such custom integrations will decrease.

Q: Can the inclusion logic be nested, such as X AND (Y OR Z)?

Yes, inclusion logic can be nested as sub-groups containing AND/OR expressions. This was not demonstrated in the webinar but is supported by the CDS Authoring Tool. To create a new nested sub-group, you can use the "indent" button in the top-right section of the first data element in the sub-group.

Q: How do the health systems and hospitals know that these artifacts are available for them to use and integrate? How many health systems or end users are accessing this information? Thanks!

AHRQ and the CDS Connect team regularly participate in and present at conferences, webinars, work groups, and meetings across the quality improvement, informatics, clinical decision support, interoperability and patient-centered research communities. We collaborate with other government agencies who are working on similar initiatives to align and enhance the impact of this work. In addition, we have released white papers, blogs, and have a live web-site that is publicly available with searchable artifacts.

We have no way of knowing how many health systems or end users have implemented logic that is publicly available on the CDS Connect Repository, however in the past year there has been 15,000 views of the 33 CDS artifacts!

Q: When using CDS hooks with a compatible EHR, what does the user interface look like? Is it what you displayed in the demo EHR environment, or is the graphical user interface customized by the institution?

CDS Hooks provides an API that specifies how EHRs can interact with 3rd party CDS services. It primarily specifies how *data* is exchanged, so most of the presentation format is determined by the EHR invoking the service. This allows the CDS response to be displayed in a style consistent with the rest of the EHR's user interface. That said, there are three ways a CDS Hooks service may specify some aspect of the presentation:

- the "details" component of the CDS response may contain minimally formatted text (using GitHub Flavored Markdown)
- returned "suggestions" are displayed as buttons (although the style of the button is up to the EHR)
- if the response contains a link to a SMART App, the SMART App developer has full control of the SMART App's user interface

During the demonstration, the CDS Hooks Sandbox was used. The CDS Hooks Sandbox presents a very generic user interface, as its main purpose is to test the general interaction between a mock EHR and a CDS service.

Q: How does this compare with OpenCDS? And how does the design of it relate with OHDSI?

The CDS Connect team is not closely involved in the OpenCDS or OHDSI efforts, although we are broadly aware of them. We recommend that you reach out to each organization if you'd like more information about their tools or wish to become involved in their efforts. That said, we'll attempt to answer the question as best we can.

OpenCDS offers several open source tools related to CDS, including CDS builders based on a JBoss Drools and JBoss jBPM. These present a very different user experience for developing CDS logic than the CDS Authoring Tool. In addition, while OpenCDS uses standard data models such as HL7 Virtual Medical Record (vMR) and HL7 FHIR, it does not appear to produce HL7 CQL to represent the CDS logic.

OHDSI has a strong focus on large-scale health data analytics, whereas the CDS Authoring Tool is focused on rules-based logic. OHDSI projects and tooling use a data model called the Observational Medical Outcomes Partnership (OMOP) Common Data Model (CDM). There appears to have been some discussion within the OHDSI community regarding the possibility of translating CQL/ELM to OMOP JSON, but it's not clear if anyone has been able to successfully achieve this.

Q: How do artifacts generated by this tool relate to eCQM artifacts for common topics.

eCQMs and CDS developed by the Authoring tool are very related. Both are expressed using Clinical Quality Language (CQL), which is an HL7 domain-specific language focused on clinical quality. Both eCQMs and CDS are useful tools to drive quality improvement: CDS to inform and drive evidence-based decisions at the point of care and eCQMs to measure compliance with evidence-based expectations. Often, portions of a CQL phrase within eCQM logic can be re-used when defining CDS logic and vice versa. Where they differ is the data model that is being used for each effort. eCQMs use the Quality Data Model and the CDS Authoring Tool uses the FHIR data model. Efforts are underway in the eCQM community to evaluate the transition to another data model, but a firm decision has not yet been made.

Q: Can actionable functions be built into the recommendation such as a button to order meds?

The CDS Authoring Tool currently supports text-based recommendation messages only. It's important to note that the CQL standard is designed to support CDS logic only, so it is not capable of specifying how a user interface should work. For example, you could build an object representing an order in CQL, but you would need to use another standard to associate that CQL expression with an "order button" in an EHR. To support this type of workflow, CQL would need to be used in conjunction with FHIR Clinical Reasoning or CDS Hooks.

One approach to using the CDS Authoring Tool with CDS Hooks is to write a CDS Hook that uses CQL expressions from the authored CDS to determine if an order should be suggested. In this case, however, the details of the order would be implemented in the CDS Hook service, not in the CQL. The CQL Services prototype that was demonstrated in the webinar can support this workflow. For an example, see the alternate statin use configuration.

Q: Because CMS requires each group or EC to select from a pre-define/vetted list of ECQM, how is creating a custom CDS standard beneficial, what do you see the Use Case being? Would this be beneficial for organization strategies outside of QPP reporting?

Electronic clinical quality measures (eCQMs) measure care retrospectively – well after the care has been provided. CDS provides an opportunity to guide decisions at the point of care, often based on the same evidence-based guidelines that have inspired and informed the development of eCQMs. By having a common "standard" (i.e., Clinical Quality Language [CQL]) to express *both* quality improvement concepts, it enables vendors and healthcare organizations to learn, integrate and implement a single standard to meet both needs. In addition, it allows authors to re-use CQL expressions between corresponding CDS artifacts and eCQMs. This re-use is not only more efficient but can also better support evidence-based decision-making that leads to higher eCQM scores. It's worth noting, however, that CMS eCQMs currently use the Quality Data Model, while much of the CDS community uses a FHIR data model. This vision of easily re-usable clinical logic can only be fully realized when eCQMs and their corresponding CDS artifacts agree upon a common data model.

CQL can be used to develop CDS that aligns with an array of evidence-based resources (e.g., HEDIS measures, clinical practice guidelines, peer-reviewed research studies), enabling developers and healthcare organizations to create CDS expressions that support other reporting requirements, organizational objectives or research. In addition, the standards-based CDS expression can be shared with other organizations and integrated in to their HIT systems far more readily than logic expressed in varied proprietary formats.

Q: What is the current regulation of CDS from FDA?

The Food and Drug Administration (FDA) has indicated through their initiatives and guidance that they intend to take a risk-based approach to software regulation, including clinical decision support (CDS), limiting their oversight and regulation. The most recent FDA guidance was issued in December, 2018, in a report on Non-Device Software Functions: Impact to Health and Best Practices, a provision of the 21st Century Cures Act. The guidance

proposes to reduce oversight of software functions that analyzes patient data and provide notification to a healthcare provider through means such as alarms or alerts.

Other relevant FDA guidance issued previously includes:

- The 2014 Food and Drug Administration Safety and Innovation Act (FDASIA) Health IT Report (in consultation with the Office of the National Coordinator for Health Information Technology (ONC) and the Federal Communications Commission (FCC)), which outlined a risk-based approach to the regulation of health information technology generally, and CDS support products specifically.
- The 21st Century "Cures Act," enacted by Congress in 2016 and formalizing the FDA's risk-based approach.
- July 2017, FDA Digital Health Innovation Action Plan, which lays out the agency's "vision for fostering digital health innovation while continuing to protect and promote the public health."
- December 8, 2017, FDA Clinical and Patient Decision Support draft guidance, updating and clarifying the agency's approach to CDS products in light of the Cures Act.
- February 15,2019, FDA Digital Health Software Precertification (Pre-Cert) Program is informing the development of a future regulatory model for oversight of software-based medical devices.

Q: Is there a list of EHR Vendors that support CQL?

Vendors are currently in varying stages of progress for their CQL support. In general, vendors have not publicly announced their roadmaps, but we can expect to start seeing support for CQL-based eCQMs soon since 2019 CMS eCQMs are being published as CQL. We recommend contacting your vendors for more details.

For more information about integrating CQL-based CDS using tools available today, please see the response to a previous question on the same topic.

Q: Is it possible to generate Recommendations dynamically? e.g. localized resources

The "Recommendations" portion of the CDS Authoring Tool allows recommendation text to be entered in any language. It does not, however, support an easy mechanism for providing messages in multiple languages that can be dynamically selected based on the operating locale.

Localization could potentially be accomplished at the integration points between the CQL logic and the EHR. For example, instead of typing a sentence into the "Recommendations" portion of the CDS Authoring Tool, an author could enter a *key* or *code* representing the message instead. When the integrating system executes the CDS logic, it receives the *key* as the recommendation and could use that key to look up a properly localized message in a resource file. Another approach might be to ignore the returned recommendation text and instead use other expressions in the CQL to determine if a message should be displayed. Then it would be up to the integrating system to display the right message in the right language based on those other CQL expression values (typically *true* / *false* values).

Q: How can you determine if the artifacts are evidence-based and appraised for quality?

The CDS Connect Repository provides numerous metadata fields that enable a CDS artifact contributor to:

- Describe the evidence-based research source from which the logic was derived and provide a hyperlink to the evidence
- List the specific recommendation statements that informed the logic, along with the Grades of Recommendation Assessment, Development and Evaluation (GRADE) and quality of the evidence
- List references that were used during the knowledge translation process and provide hyperlinks to the articles and resources
- List the names of CDS developers, along with their organizational affiliation and hyperlinks to their professional credentials
- List Cautions and Limitations regarding the CDS expression

- List Testing and Pilot Experiences related to the artifact to convey the effectiveness, impact, accuracy and usage of the logic
- Specify the Status of the artifact (e.g., Active, Inactive, Draft). Many of the CDS Connect artifacts are in Draft status (i.e., they may not have been tested in a clinical environment).

An artifact viewer can evaluate metadata responses to determine their impression of the artifact's quality, maturity and degree of alignment with the evidence-based source. Ultimately, they are the judge of whether the artifact is a good fit for their organization or if they prefer to tweak the expression prior to implementing it.

When artifacts are presented for publishing on the CDS Connect Repository, project staff perform a high level technical and clinical review of the materials and metadata responses to check for consistencies across the evidence, description, semi-structured and structured code. Any inconsistencies are rectified prior to publishing the work.

Q: Are there any analytics based on the CDS displaying and additional follow-up items in additional encounters?

CDS Connect contributors are encouraged to include information about how the CDS expression performed during clinical testing and implementation within their artifact entry. If the artifact was implemented in a clinical environment and the contributor chooses to share analytic results of the implementation, the information will likely be in the Testing Experience section of the Repository. The level of detail will vary by contributor. This link will take you to CDS developed by the CDS Connect team last year to support pain management decision making. Details regarding the impact, accuracy and usage of the CDS as well as the lessons learned during the implementation process are outlined in the Testing Experience section.

Q: Can individual artifacts be prioritized in any way so that when several "succeed" a system can select which to display?

Authors can provide several potential recommendations within a single artifact in the CDS Authoring Tool. If multiple recommendations apply to the patient, only the *first* one will be returned. In this way, authors use the order of the recommendations within an artifact to prioritize them.

In the case of several *different* artifacts all triggering recommendations at the same time, there isn't a standard way to indicate which should be prioritized unless you wrap them all in a single encompassing artifact with logic to indicate the prioritization rules.

Q: How can CDS artifacts be downloaded and edited to add or modify inclusions, exclusions, and other parameters?

There are two main online components to the CDS Connect project:

- CDS Connect Repository: for finding, downloading, and sharing artifacts
- CDS Authoring Tool: for creating CDS logic

Currently, they are not linked in any automated way. If an you want to share CDS logic you created in the CDS Authoring Tool, you need export it as CQL (a zip file), create a new artifact in the CDS Connect Repository, and upload your CQL to that artifact in the repository. This is a multi-step process that we may streamline in the future. If you're interested in sharing CDS on the repository, please reach out via the "Contact Us" link on the repository.

As for someone else adding additional inclusion and exclusion changes, parameters, etc. – they are free to download CQL shared on the repository and modify it. As of now, all the artifacts in the repository are released using open licenses. It is not currently possible to re-import the CDS logic into the CDS Authoring Tool to make changes – so any changes would need to be done by hand. If you're hoping to share CDS in the Authoring Tool so that other people can make their own copies and modifications within the tool, that's not currently supported, but we'll consider it for future development.

Q: Is the CDS Authoring Tool open source? How does it compare with other open source tools for CDS?

The CDS Authoring Tool is open source and available on GitHub at: https://github.com/AHRQ-CDS/AHRQ-CDS-Connect-Authoring-Tool

In addition, CDS Connect offers several other open source tools on GitHub here: https://github.com/AHRQ-CDS

As far as we're aware, the CDS Authoring Tool is unique as an open source web-based tool for creating CQL-based CDS. That said, there are a few other open source tools that are related in some ways:

- Measure Authoring Tool (MAT): The MAT is an online tool that creates CQL, but it's only for eCQMs and it targets QDM instead of FHIR. The MAT assumes its users are very familiar with the CQL standard.
- KNARTwork: KNARTwork is also an online editor, but it focuses more on metadata-related standards and requires logic to be pasted in as ELM XML (so you must author the CQL elsewhere).
- CQL Runner: CQL Runner is a great online tool for experimenting with CQL, but it's aimed more at testing CQL expressions than developing full artifacts; you type well-formed CQL into an input box, it colorizes it for you, and allows you to run it against a FHIR server.

If you're aware of other CQL-based CDS authoring tools, we would love to hear about them!