CIMI and FHIR Tooling Road-Map

CIMI and Cognitive Medicine graciously sponsored the first ever CIMI / FHIR Tooling Connectathon on September 6th-8th in San Diego. The goal was to bring together all the key software engineers involved in various open-source model authoring projects related to CIMI and FHIR. Our task was to provide a proposed road-map for the larger CIMI team to discuss and consider during the HL7 Plenary meeting the following week. This document should be considered our initial ‘proposal’, and should not be considered ‘official’ until reviewed and approved by the full workgroup.

The real-world road-map involves both short-term tactical tools that offer a quick option to deliver a critical-path functionality as a proof of concept (POC) which can elevate critical challenges and provide practical solutions. These tactical efforts would often then be integrated together into more robustly architected long-term solution.

HL7 January Ballot-cycle Targets

The entire group will be giving weekly report-outs on each ongoing projects at the Thursday CIMI call. But the call on October 5th will be our ‘drop dead’ cut-off for final decisions on which projects will make the cut-off for the January ballot cycle.

Projects tentatively targeted for the January cycle:

- CIMI to FHIR profile publishing
- Centralized Transformation, Validation and Publishing Infrastructure
- Java based AOM Library with I/O to BMM & ADL
- CIMI-LE documentation, gap analysis and ANTLR grammar
- Consolidated Mapping language for producing FHIR Profiles from Models
- Java Code-Generator for FHIR Profile Extensions
- CIMI Model Repository and Usage Registry

Ongoing research related to CIMI Tooling:

- Research integrating MDHT with KOMET/ISAAC
- Research using CEDAR for CIMI Model development
- Research challenges mixing Netbeans and Eclipse Modeling Framework

CIMI to FHIR profile publishing

The CIMI (BMM/ADL) to FHIR profile publishing tool is a good example of the POC strategy. Claud intends to build a stand-alone tool for the January 2018 ballot cycle. This

Java based AOM Library with I/O to BMM & ADL
The central capability for a tooling stack for BMM / ADL is the ability to serialize and deserialize
the files to and from an in-memory representation. This in-memory representation is known as
Archetype Object Model (AOM). Currently the only AOM implementation is written in Eiffel, so a
Java implementation needs to be written. Claude will be working to build the Java
implementation and then Richard and Kurt will provide a C# / .Net version as well.

Over-time, the AOM will be used as the ‘core hub’ for a variety of modeling and transformation
tasks. Numerous different import, export, transform and validate capabilities will be built around
the AOM core. Any modeling syntax that can be imported into the AOM, could then be
validated, and then serialized into any other AOM compliant syntax.

Our mid-term strategy is to build a pluggable architecture that allows for a family of modeling
semantics, each of which shared set of modeling capabilities, yet each is allowed to explore
alternative methodologies to support different authoring communities. Some authors will prefer
a compact Domain Specific Language, which is authored through a classic text-editing
approach. Other authoring communities find text-editing terrifying, and will prefer a drag-n-drop
tool with prompting for details, which feels more supportive of less technically experienced
authors. But the critical challenge is being able to bi-directionally transform losslessly between
the compliant semantic capabilities.

Syntaxes targeted for Import, Validation or Export to / from the AOM:

- BMM / ADL - the primary serialization format for CIMI
- CEML - Import and Export
- Cameo - Import and Export
- FHIR Profiles / Structure-Defs - Export
- CQL Model-Info files - Export

CIMI-LE - documentation, gap analysis and ANTLR grammar

CEML and Cameo are two, very similar, Domain Specific Languages (DSLs) for representing
clinical models in a form that is more amenable to human authoring and readability. The CIMI-
LE effort will work to harmonize those two syntaxes to determine if a single fully aligned syntax
could be achieved, and if not, then determine where gaps exist.

The goal of the project is to offer alternate methodologies for authoring clinical models, whereby
different needs and preferences can be accommodated, while still producing 100% compatible
models. Teams that prefer to directly write a text based language, could choose CIMI-LE, and
those that prefer drag-n-drop with prompts for authoring could go that route. The critical issue is
that the models produced would be identical in every way when serialized as BMM/ADL within
the repository and when published.

Consolidated Mapping language for producing FHIR Profiles from Models
It was discovered that the mapping languages developed by the CEML and Cameo teams for transforming clinical models into FHIR profiles is very nearly identical. Therefore it has been proposed that those languages be coalesced into a single syntax and then used across all FHIR transformation tasks.

Joey, Nathan and Mark will be documenting the similarities and differences and proposing a unified syntax. An ANTLR grammar will then be produced.

**Java Code-Generator for FHIR Profile Extensions**

Dealing with the ugliness of writing code against profiles that contain extensions is excruciatingly painful. Roman, Bryn and James will be working to build a tool that will automatically produce Java classes with getters and setters that allow for all extensions to be access as first class objects. This dramatically reduces the pain of Extensions.

**CIMI Model Repository and Usage Registry**

Model and profile adoption is a critical factor that can drive recognition of success. Craig has been developing a tool that can track the deployment of Models and Profiles and publish the results.