A Method for Applying Graphical Templates to HL7 CDA

Rik Smithies
Chair HL7 UK
NProgram Ltd.

I. Introduction

The English National Programme for IT (NPfIT) is being delivered by NHS Connecting for Health (NHS CFH). Initial NPfIT work built upon earlier local HL7 V3 specifications and concentrated on constraining and refining the RIM in a bottom up approach. Models were built from RIM components assembled into messages and CMETs (Common Message Element Types). These artifacts were created with the HL7 modeling tools based on Microsoft Visio. CMETs were reused across message domains and message definitions were constructed by piecing CMETs together. This approach was successful, with several live systems in operation and many millions of messages having been sent.

Initial CFH work concentrated on a transactional style of interaction, with machine consumption of message content. Examples are a national patient demographics system, electronic medication prescriptions, and lossless transfer of fully-structured and coded General Practice records.

Subsequent phases involved a more document based style of exchange, and adopting the HL7 Clinical Document Architecture[1] seemed appropriate. The challenge was to transition an operational message development system, with an existing set of reusable artifacts, into a CDA compatible methodology

II. Model Development Styles

A. CDA Methodology

CDA involves what might be called a top down methodology, starting from a complete "off the shelf" model. There is only one CDA model and one XML schema for it. This has the advantage of consistent XML element names in every CDA document instance. A single XSLT stylesheet can display all CDA documents, giving great portability.

CDA implementation guides typically constrain the model with statements about restrictions applying in that context. These may be codes that are allowable, excluding others, or classes in the model that are not to be used, or have restricted cardinality. Such implementation guide constraints may be in document style narrative text, which are easy for non-technical staff to produce. They may also be in a carefully structured text that can be machine translated into XPATH assertions (Schematron), which is how strict machine enforceable checking works in many CDA implementations

While this is an effective and powerful methodology, it differs from other V3 modeling techniques.

B. Other V3 Models

Most non-CDA HL7 modeling employs diagrammatic constraints. Typically, a new HL7 V3 model is drawn onto an initially blank sheet by adding RIM components and attaching CMETs. The CMETs themselves have been previously drawn in the same way. Although the process is one of drawing shapes and assembling models, the underlying method and result is always actually one of constraining and refining down. If a model, such as the RIM itself or a specialized domain model, allows multiple instances of a class then this will be shown as one class with a "many" cardinality (eg. 0..*). A derived model can be drawn with two instances of that class, having different names ("clone names"). This can appear to be an extension and a building up process, but it is always truly a restriction of flexibility, hence the common term "constraint".

Any V3 model can be constrained by drawing a more restricted model, though, as described, this may actually be diagrammatically expanded. A single repeatable class in the parent model may turn into several in the restricted child. It is also possible to constrain with non-graphical information such as documentary comments and tables. In fact virtually all diagram based constraints also have some textual restrictions. CDA however, chooses to disallow restriction of the model by derivation of a new constrained model. Such a constraint would be legal HL7 V3, but is not in scope of what CDA allows. There are good reasons for this, mentioned previously, such as preventing the use of variable clone names that result in incompatible XML instances.

In practical terms this means that CDA cannot be constrained or extended with CMETs defined elsewhere, because there is nowhere to plug them into the model.

III. A Combined Methodology

Models that constrain and refine others normally have different XML element names from the originals. CDA has a single constant XML representation that rules out direct customization of the model itself.

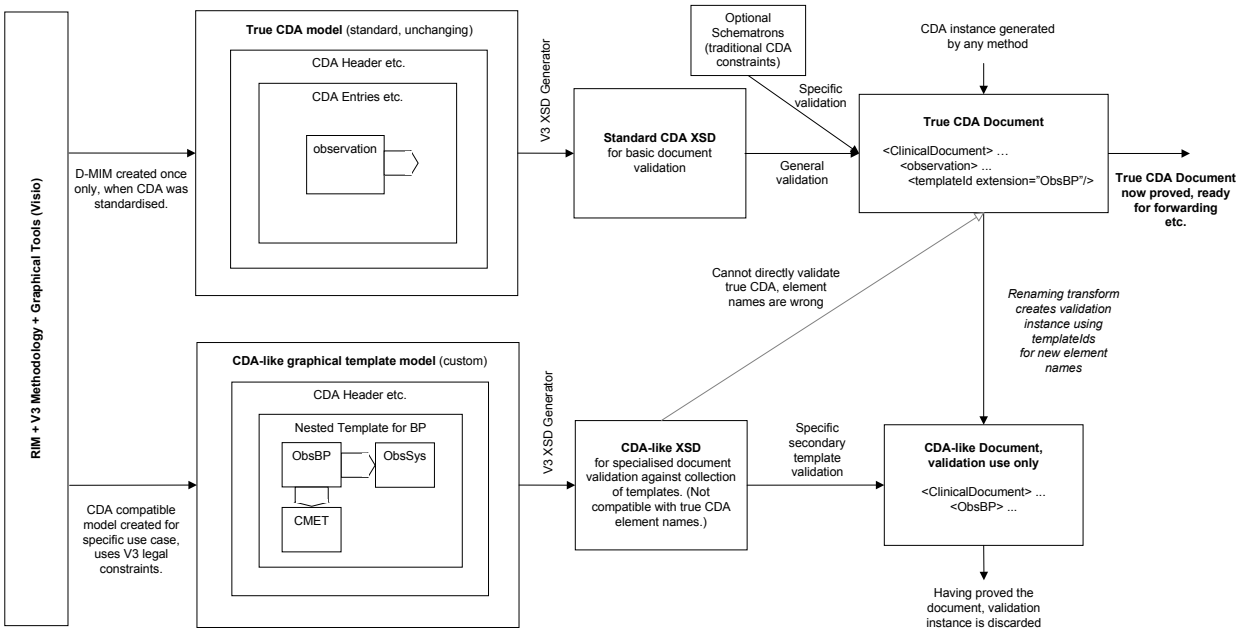

Figure 1. Outline of the method

A special method is therefore needed to allow graphical constraints like CMETs to be used with CDA. Graphical constraints are well supported by the HL7 tools, including generation of XSD for checking conformance of messages to models. The diagrams also fit well with common implementation guide styles, such as tabular attribute level documentation. This is desirable in some environments where there is also an investment in non-CDA HL7 version 3.

The new methodology uses CMETs and other model fragments in an indirect manner. Rather than connecting the CMET and CDA together directly, making one large model, the CMET can instead be used as an HL7 template[2]. Realizing that there is no difference between a CMET, or other R-MIM, and a template is key to understanding the technique. The CMET/template is applied to CDA as a secondary model. In other words the final model is made of several disconnected sub-models, each of which defines some part of the whole. A valid CDA document for these constraints will conform to the combination of the separate models. The templates thus provide an extra level of modeling over the underlying CDA, allowing piece by piece refinement, in a diagrammatic fashion similar to regular V3 modeling. Clearly for this to work the various models involved need to be compatible with each other and not contradictory. All templates need to be valid refinements of the relevant part of CDA itself. (This can be achieved through careful modeling according to HL7 rules, but there is no automatic tool to ensure this relationship. Valid example CDA message instances will generally fail to validate against invalid templates, so errors can be diagnosed that way.)

IV. The Methodology Applied

Techniques were developed at CFH to associate such templates with CDA models and XML instances, so that detailed XSD checking of document instances is possible (see Fig. 1). A validation only "CDA-like" master template model can be assembled from several CMETs/templates. This model can use HL7 choice boxes to show the templates allowed at each point, and a schema can be generated in the normal way. Using the templateId element, document instances are annotated with the appropriate template while still conforming to true vanilla CDA structures. The annotations allow a simple transform to automatically create a secondary XML file that is checkable against the templated schema.

The methods developed are technically straightforward and largely tooling independent. CFH has its own set of customized message development tools, but little of this is necessary to support the method, which can equally well be used with the standard HL7 toolset.

V. Conclusion

The work described showed that it is possible to use conventional V3 modeling techniques with CDA, as an alternative to, or in conjunction with, more commonly used documentary style constraints.

Each approach has its advantages and together they provide greater flexibility to help adoption of CDA.

Acknowledgements

The author thanks those at NHS CFH and in the HL7 community that helped foster the ideas presented in this paper.

References

[1] HL7 Clinical Document Architecture Release 2, http://www.hl7.org/v3ballot/html/infrastructure/cda/cda.htm
[2] HL7 Specification and Use of Reusable Constraint Templates, http://www.hl7.org/v3ballot/html/infrastructure/templates/templates.htm