A Method for Applying Graphical Templates to HL7 CDA

Rik Smithies
Chair HL7 UK
NProgram Ltd, NHS CFH
Introduction

• The English National Programme for IT (NPfIT)

• A project being delivered by:
  **NHS Connecting for Health**
Message Design

• First NHS CFH messages designed using traditional HL7 V3 process
• Later, a need to produce more document-like messages
  – CDA the obvious choice
  – Good fit with existing V3 messages and processes
  – Wanted to re-use components and investment in staff knowledge and tools
Message Design Process

• “Traditional” V3 methodology followed
• R-MIMs (“Visios”) and CMETs created, shared, re-used
• Implementation guides produced
  – 30+ iterations over several years
  – Using graphical layout
  – Look and feel, as well as procedure, became familiar to CFH and to suppliers
  – Not unlike the HL7 Ballot Pack (no coincidence!)
HL7 Modelling Overview

- V3 is reference model based

The 6 RIM backbone classes
Modelling Principles

• V3 models are built up directly from RIM components
  – and from CMETs (common elements)
  – which are themselves built from the RIM

• Although this is a bottom-up construction process, the underlying method is always one of a series of restrictions downwards from the RIM

• Everything is a “constraint”
Modelling Example

Note there are now more participations even though the model is a constraint on the original. Attributes are unconstrained, in clone classes as 1..1 is more constrained.

Attributes now more constrained, in clone classes from the RIM.

Original was 0..*. 3 of 1..1 is more constrained.

Multiplicities of relationship is unconstrained, zero to many.
“Unrolling” and “Constraint”

• Previous slide showed that models can be “unrolled” to derive new models.
• In effect, pieces can be added on.
  – New “clone” classes can be added
• This is consistent with continual constraint downwards.
• There is actually little freedom to change things - as it should be.
CDA

- CDA is an HL7 V3 model (and supporting methodology) that has been developed from the RIM as described.
- Importantly, it is designed to be used as is, with no further derivation at model level.
- The model is always constrained for the specific use case, but “cloning” to add new classes is not allowed.
- This is a more strict approach than other V3 models.
CDA Methodology

- CDA implementation guides document how the existing classes are to be used.
- No new classes are derived, or unrolled.
- This gives consistent element names in the XML.
- Hence a single XSLT stylesheet can display all CDA documents.
- This gives the greatest portability of CDA documents.
CDA Implementation Guides

• Guides constrain the model with statements about restrictions for that context
• eg. “Use only these codes”, “don’t use this class”
• Guides tend to have a narrative, documentary format
• Easy for non-technical staff to produce
• Can make use of compilable “structured text”, giving ability to machine check the rules
• A powerful and successful approach, but nevertheless different from other V3 techniques
Graphical Methodology

• “Graphical” constraints are well supported by V3 tools (Visio etc.)
• Can draw RIM-based models, generate XSD and documentation in a common format.
• Can include “library” CMETs defined elsewhere
Differing Processes

• Models that constrain and refine others have different XML names from the original
• Not allowed in CDA
• Differing tools and philosophies are inconvenient for organisations using both CDA and non-CDA V3 messaging.
A Combined Methodology

- CDA cannot be re-modelled directly
- An indirect approach is needed
- HL7 Templates are the key
- “Template” is an overused word in the software world…
- An HL7 Template is an additional constraint to be used with another model
- A layering is implied
Ways to extend a model

Looping arrow allows extending this model here
Extended in-place
Another way to extend this...
...create another model

eg. "systolic"

eg. "BloodPressure"

eg. "diastolic"

eg. "position"
Complete model is 2 models - master and template.

Original model is left unchanged (could be CDA)

Extra modelling done in parallel
Secondary Model

- The resulting model is a combination of the two models.
- This is not dissimilar to constraining CDA with textual statements. A secondary model is being described.
- Documents/messages must conform to both models.
- The secondary model (template) must naturally be compatible with the first.
Template Class Names

• The template will have some non-CDA class names.
• It may have CDA “Observations” renamed to “systolicBP” and “diastolicBP”
• XSD generated will not match CDA and so cannot be used directly
• In fact the message elements must conform to 2 classes at once – the inherent CDA class, and the template class.
Element Names

- Validating against 2 models implies knowing which extra model applies at each point of the XML.
- A simple method is to use the “templateld” HL7 attribute in the document.
- Templateld acts as like a secondary class name in the instance. It is allowed on any class.
- “This class (also) conforms to template class X”
Two Stage Validation

• A 2 stage modelling is followed by a 2 stage validation

• Class instances have 2 names. The normal XML name, plus another in the templateId.

• An XSLT transform can rename the instance so the templateId ("systolicBP") is put into to the actual class name
Example Transformation

Original document:

```xml
<ClinicalDocument>
  <Observation>
    <templateId extension="MyTemplate.systolicBP"> <!-- redundant name -->
    <code displayName="Systolic Blood Pressure"/>
  </Observation>
</ClinicalDocument>
```

becomes, after transformation into a validation-only document:

```xml
<ClinicalDocument>
  <systolicBP>
    <code displayName="Systolic Blood Pressure"/>
  </systolicBP>
</ClinicalDocument>
```

This will then validate against a schema file that includes “MyTemplate.xsd”.
Validation CDA Model

- XSD validation works on a whole document, not a fragment.
- A master CDA-like model needs to be created.
- CDA is re-drawn in Visio
- The templates are inserted at the appropriate positions, by using them as CMETs.
- This model is what CDA would be like if it could be extended directly.
- The resulting single XSD can validate a true CDA instance once it has been “name transformed”.

Method Summary

- All document instances are true CDA
- CDA constraints are modelled in Visio as graphical templates
- These are for documentation and for validation
- Templates are assembled into a new CDA-like validation model (in Visio)
- “templatelds” say which templates are being used
- The document is still true CDA
  - will validate against the CDA schema
  - will display with the CDA stylesheet
  - can conform to any other written CDA implementation guide
- As an optional step, a template aware application can transform the document and validate against the templates
Conclusion

- Familiar graphical techniques can work with CDA
- The key is a 2 level modelling, using Templates
- The method involves only existing HL7 tools and a simple stylesheet
- This method has been put to use in the UK and is the standard way that CDA is modelled in NHS Connecting for Health

• *Thanks to those at NHS CFH and HL7 who helped develop these methods.*