Archetypes for HL7 CDA Documents

HL7 CDA has been chosen by a number of organisations and communities as a standard to underpin clinical information exchange. This decision has resulted in a number of significant implementation challenges for those wishing to standardise and prescribe the content of specific document types, such as hospital discharge summaries, referrals, specialist reports, etc. A number of significant problems result directly from the way the computable entries in CDA documents are currently represented.

openEHR offers a way to overcome these problems. This discussion paper outlines the problems inherent in the conventional CDA approach and a potential solution to these problems based on openEHR Archetypes.

keywords: HL7, RIM, openEHR, archetypes, templates, CDA, SNOMED, clinical

Goals of CDA

HL7 Clinical Document Architecture (CDA) is a specification for the creation of electronic documents such as discharge summaries and referrals, marked up using XML, such that they can be processed simply and reliably for human interpretation whilst also supporting more extensive computational processing. The former, human readable portion can be attested by a clinician and rendered in a browser using a standard XSL stylesheet. The latter is achieved through coded data structures and terminologies based on HL7's Reference Information Model (RIM). CDA Release 2, specifies a format whereby each section of the electronic document can contain a number of computable "entries", each of which adheres to HL7's RIM-based Clinical Statement pattern[1]. Many proponents believe that these coded entries enable semantic interoperability amongst systems sharing CDA documents and will lead to improved decision support and possibly allow CDA documents to be the foundation of electronic health records.

Others have strong doubts about the promise of CDA. According to the primary authors of the specification [DOL2005] - "While CDA R2 doesn't fully enable plug and play semantic interoperability, it takes us yet another step closer." Using the HL7 RIM as the underpinning specification for coded entries brings significant problems in making even small steps realizable, and brings into question the very goals of CDA R2.

Problems

Most of CDA's primary shortcomings arise from the requirement to use RIM-based modelling for representing detailed clinical content in each CDA *Entry*. Four significant problems are:-

- Lack of Text <--> coded Entry cohesion
- Inconsistent representation of clinical concepts
- Lack of specialisation semantics
- Lack of tools or methodology to assist clinicians design, evaluate and standardise detailed clinical models.

Let's examine each of these in some detail.

---

[1] CDA R2 has it's own variant of this pattern, giving a choice of Encounter, Observation, ObservationMedia, Organizer, RegionOfInterest, SubstanceAdministration, Supply, Procedure or generic Act acts.

Text - coded Entry cohesion

In CDA documents, the narrative or text block in each section is the component that is attested by the author (or author's proxy) and displayable consistently in a browser. For instance, the Continuity of Care Document specifies a section for "Vital Signs" , containing a a text table of, and one or more Entries for weight, height, systolic BP, diastolic BP observations.

Date / Time	Nov 14, 1999	April 7, 2000
Height	177 cm	177 cm
Weight	86 kg	88 kg
Blood Pressure	132/86 mmHg	145/88 mmHg

The computer-processable "coded" entries are optional and considered only "nice-to-have". Where coded Entries are placed in the document, they often abstract away the language of the clinician used in the narrative section. Sometimes the narrative abstracts away specific language used in the coded Entries. **The RIM has no mechanism to adequately tie the narrative to the entry(ies), let alone to allow transformation from one form to the other!**

Inconsistent representation of clinical concepts

The HL7 RIM uses a combination of structural components and coded terms to represent clinical concepts. These are variously: classes (e.g. the *Observation* act); class attributes (e.g. *Observation.code*); referenced terminology coded terms (e.g. LOINC code *4548-4 Hemoglobin A1C*); datatypes (e.g. CD datatype for holding potentially post-coordinated compound term expressions); auxiliary classes, related through Act-relationships (e.g. component Observation classes). Moreover, the way these are assembled results in inconsistencies in data representation. There is no consistent way to express constraints. There is an incomplete draft standard for HL7 Templates, but there is no expectation that receiving systems will have access to or use the same Templates used by the sending system, since the RIM + vocabularies are supposed to be the unequivocal source of all semantics.

Consider a fragment of a CDA template for Barthel Index[2], which might look something like that of Figure 1. and where what is being recorded is an artificial recording concept, unlike a person's height. Barthel's index is a compound assessment with a total score. The integer value (the total score of the 10 constituent functional assessments) is meaningless on its own. The RIM requires that the result be inconsistently spread across a range of different attributes, classes, potential vocabularies etc. One piece in *Observation.value*, another in *Observation.InterpretationCode*, yet another in *Observation.derivationExpr*, some more in component *Observation.value*, but with the meaning and relative weighting of those values lost to all and sundry.

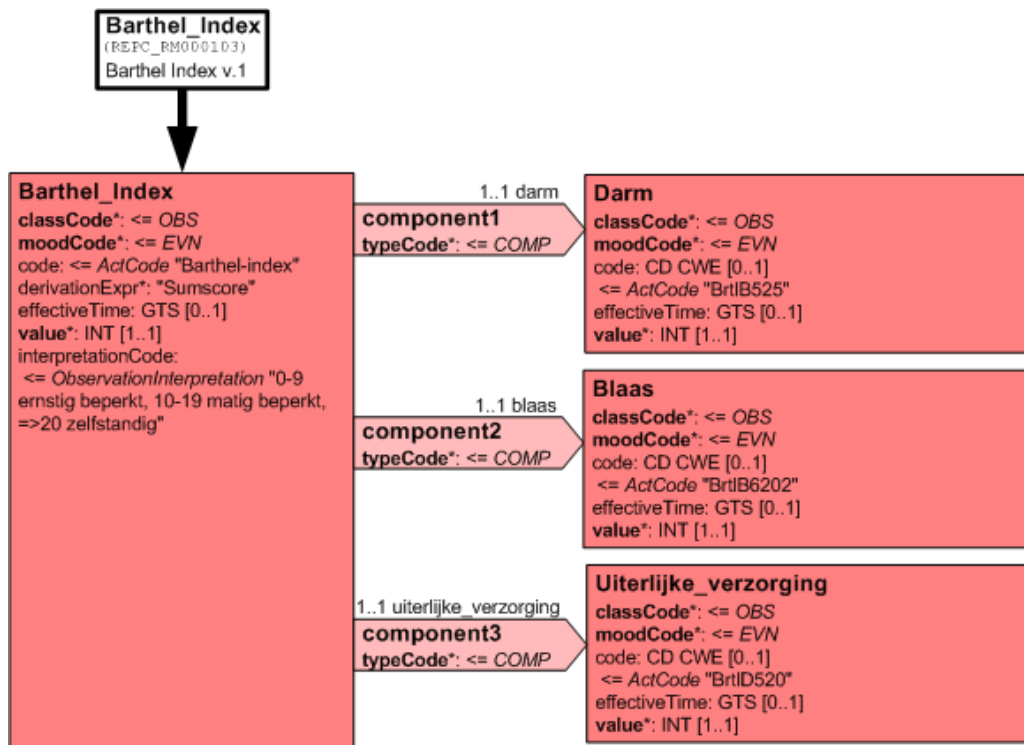[2] a measure of a person's functional capability in performing common daily tasks.

Figure 1: implicit matching of Observation's code and value

For so many observations, the HL7 RIM has the code/value dilemma. The *Observation.code* implies a specification for what is being observed. The *Observation.value* should be the result of the observation. The value should accord with the specification. **The RIM has no mechanism to ensure this**!

Lack of specialisation semantics

It is quite clear that some types of observations, assessments, procedures, etc. are specialisations of more generic ones. However, HL7 does not have a systematic methodology for expressing specialisation in general, nor of CDA Entry types in particular. In some cases, specialisation is hardwired into the RIM classes (e.g. the *Observation* class is a specialisation of a generic Act class); in some cases specialisation is expressed through 'levels' of HL7 structural codes (e.g. a *falls risk assessment instrument task* Act is a specialisation of a *risk assessment instrument task* Act. Or a *Diner's Card* Act is a specialisation of a *CreditCard* Act. And an *umbrella liability insurance policy* Act is a special type of *ActInsuranceTypeCode* Act), where these ActCode levels can be Abstract, Specializable or Leaf; in some cases specialisation remains implicit, but might be inferred through external terminology hierarchies such as SNOMED, when used to populate *Observation.code*; in some cases one complete RMIM might be a specialisation of a DMIM or another RMIM; in some cases one LIM might be derived in some fashion from another LIM. In some cases one DAM might be derived in some fashion from another DAM. And so on it goes. **The RIM has no consistent mechanism to support specialisation semantics!**

Lack of tools or methodology

Although HL7 has an extensive methodology and a number of tools to support the production of RIM-based models, XML schemas, message specifications etc., it falls down dramatically when it comes to supporting the design, communication and agreement of detailed clinical models. The situation will only become worse, in the near term, as the HL7 organisation tries to embrace more and more coded vocabularies and whilst other organisations such as IHE

bring in profiles to try to lock down specifications for particular purposes, resulting in a proliferation of alternate specifications based on local requirements. Anyone needing to design, understand, communicate or use any of the various CDA related specification artefacts needs to be extensively versed in much of the HL7 RIM's architecture, language and undocumented usage practices. To become so versed takes years, for all but the most exceptional learners. **The RIM has no associated tooling to facilitate easy CDA document design or specification communication!**

The above problems all severely compromise the adoption of HL7 CDA as a widespread solution for clinical information exchange.

History

To many observers, the problems outlined above are all too familiar from the days of HL7 version 2. Version 2 has been extensively criticised for not having a good method for presentation of messages to humans. It has been extensively criticised for inconsistent representation of clinical concepts - "If you have seen one HL7 v2 message, you have seen exactly one HL7 v2 message". Version 2 has been criticised for lacking specialisation semantics. And Version 2 has been extensively criticised for lack of tooling and for an arcane specification representation that only trained experts can follow.

HL7 v3 was supposed to be a leap forward. Many observers watched the RIM evolve and hoped that the V3 world would herald a new era in semantic interoperability. The complexity, however was daunting and has slowed uptake dramatically. When HL7 CDA emerged it was seen as a breath of fresh air, primarily brought about by it's low barrier to entry. Bob Dolan et al [DOL2005] noted:

"Many will choose to use CDA R2 in its easy-to-implement form of just a header wrapping a non-XML body, or of a header with sections containing only narrative and no entries. This will serve to bring a lot of clinical documents into a standard format. While it may be a small step, it is then possible to incrementally add structured entries. CDA R2 offers implementers the ability to use the standard now, while over time adding sophistication."

Attractive as this may seem, it has a hidden danger. It entrenches a document model that raises expectations beyond the capability of the model and the clinical systems to deliver.

Given the four problems outlined above and the significant interest in HL7 CDA, it is time to see if the situation can be rescued.

The genesis of a solution

If we consider the HL7 RIM's Observation class as a generic framework for describing clinical observations, we note that each observation is characterised by two primary attributes. The first is the *Observation.code*, which describes the nature of the observation, i.e. what has been (or should, might, could, will be, etc.) recorded, e.g. "blood pressure". The second is the *Observation.value*, which contains the result of the observation, e.g. "132/86 mmHg". As alluded to earlier, the *Observation.code* is implicitly a specification for what is being observed. It uses HL7's CD datatype to record this implied specification, possibly as a SNOMED CT code or post-coordinated expression pointing to the concept for say "blood pressure". What we would **ideally** like is for the *Observation.code* **not** to point to "blood pressure", but to point to something like "this observation is recording systemic arterial blood pressure as two separate quantities with their magnitudes and units - the first being systolic BP, the second being diastolic BP." And we would like this specification in a computable format. Instead,

CDA implementations traditionally plonk a code from some favourite terminology into the *Observation.code* attribute, and hope that the *Observation.value* corresponds to the implementer's notion of what constitutes a valid blood pressure representation!
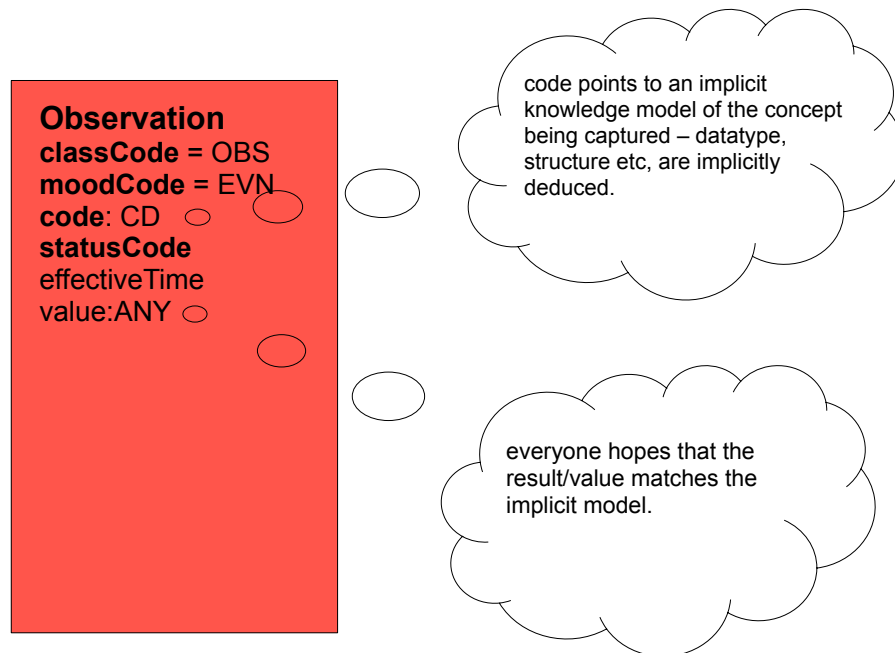

Figure 2: implicit matching of Observation's code and value

[HL7 doesn't actually have a datatype to store a pair of component parts "130 mm Hg and 85mm Hg", so an extra pair of Observation classes and Act-Relationship objects are normally added.] Sometimes these component Observations gain so much prominence that the parent Observation is discarded, as witnessed with the ASTM/HL7 Continuity of Care Document, which has no Observation for "*blood pressure"* - but two separate, uncorrelated Observations for "*Systolic blood pressure*" and "*Diastolic blood pressure*"!

Now, instead of this chaotic approach, we could replace each implementer's favourite terminology code for blood pressure with a reference to one agreed formal, computable specification for what is to be **recorded** - an *open*EHR archetype, e.g for blood pressure! By doing this, we make the specification for the value **explicit** rather than **implicit**, as illustrated in figure 3.
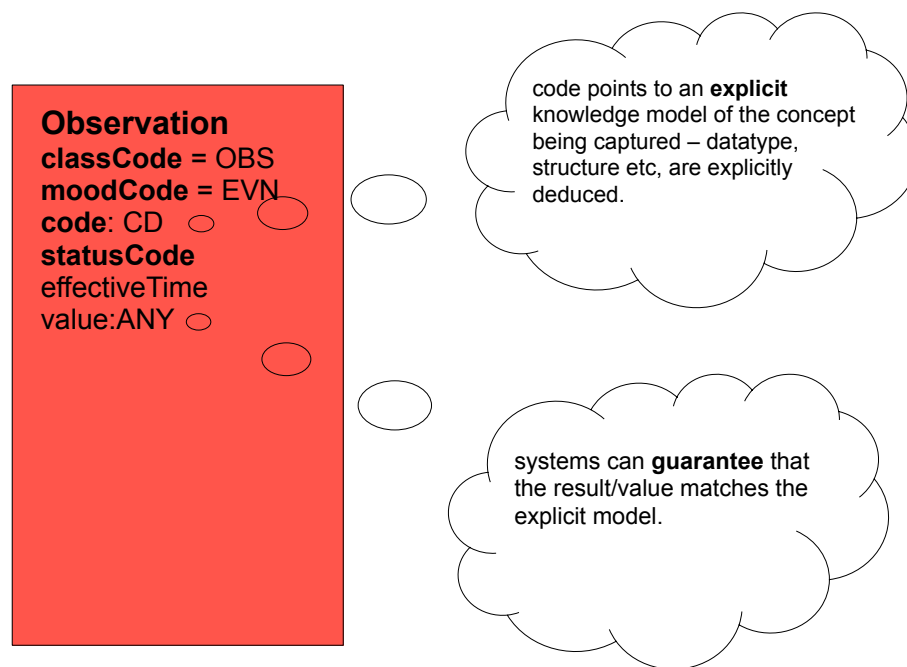
Figure 3: explicit matching of Observation's code and value

If we extrapolate this notion to all CDA entries, then each entry could use HL7's Observation Act structure to record firstly, a reference to the archetype (a computable specification for what information is captured in this entry) - stored in *Observation.code*, and secondly, the recorded data itself - stored in *Observation.value*. The data conforms to the archetype. In effect, the HL7 Observation class becomes a container for recording archetyped data - i.e it represents an observation, by a system, of a clinical observation, evaluation, instruction or action carried out by healthcare provider, other individual or device. This fits the HL7 definition of an *Observation* Act as "An Act of recognizing and noting information about the subject, and whose immediate and primary outcome (post-condition) is new data about a subject". This solution brings consistency in the way all entries are structured. The specification of what has been recorded is referenced in the *Observation.code*. The corresponding data is stored in *Observation.value*. One consistent representation for all entry types.

SNOMED Codes for Archetypes

In order to reference an archetype in a CDA Entry, we need some form of identifier. This could be done using an OID as is done with CDA TemplateIDs. It could be done by using openEHR's ArchetypeID. There are several implementation considerations that suggest a third approach - namely a SNOMED CT identifier. Consider the following:-

- The governance, management, distribution etc. of a large number of coding systems is highly problematic. It is easier for implementers to incorporate and manage updates to one, or at best a few well designed terminologies than a raft of differently structured code systems.
- Where numerous coding schemes are used, the overlap, lack of cohesion, lack of correlation between these coding systems dramatically degrades the value of clinical data, because it so impedes decision support, clinical guideline interaction and EHR querying. An 'M' for 'male', in some part of a document cannot readily be used to

correlate with diseases, morphologies, anatomical structures pertaining to males, yet described by some completely different code for "male" in another part of the document, or with yet another code for male in a computable clinical guideline. How could clinicians or researchers ever reliably construct queries if they had to reference different coding systems for different parts of the document?

- SNOMED CT predominantly contains concepts for real entities such as diseases, organisms, substances etc. It also contains some information recording artefacts[3], including, for instance, a concept termed "Clinical statement entry", below which archetyped entry concepts could be placed, as illustrated in Figure 4.

Observation
classCode = OBS
moodCode = EVN
code: CD
statusCode
effectiveTime
value:ANY

code="371911009"
codeSystem="2.16.840.1.113883.6.96"
codeSystemName="**SNOMED CT**"
displayName="**blood pressure recording**"
qualifier
  name code="xxxxxxxxxxxxx"
    displayName="**clinical concept pattern type**"
  value code="xxxxxxxxxxxxx"   displayName="**archetype**"
  qualifier
    name displayName="**archetype ID**"
    value code="xxxxxxxxxxxx"
      displayName="**openEHR-EHR-OBSERVATION.blood_pressure.v1**"
  qualifier
    name displayName="**Reference Model**"
    value code="xxxxxxxxxxxx"
      displayName="**EHR**"

"systolic"=130 mm Hg
"diastolic"= 85 mm Hg   serialised as XML
"position"="sitting"

data constrained and formatted
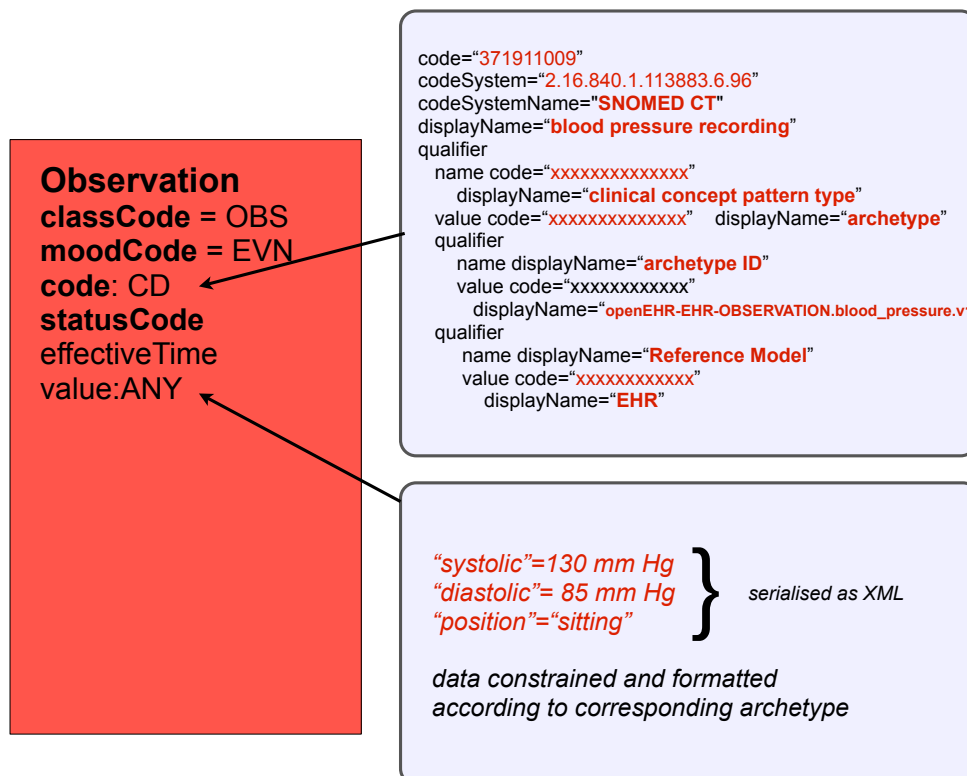according to corresponding archetype

*Figure 4: Blood pressure observation recorded using SNOMED -
registered Archetype*

SNOMED relationships would allow an archetype (i.e. a *record artefact*) to be associated with a corresponding real entity elsewhere in SNOMED (e.g. *systemic blood pressure*), thus allowing for improved searching of an archetype repository.

Archetyped data

The current HL7 CDA Release 2 specification allows the value of an observation to be of type *ANY*. By using HL7's *Encapsulated Data (ED)* datatype, with relevant MIME type, the data for the entry could be variously encoded as XML or other serialised formats as appropriate. Since the archetype node labels (via ADL *at* codes) use language familiar to clinicians, each CDA Entry's corresponding text or narrative blocks can be reliably autogenerated from the archetypes and the archetyped data. This is of significant advantage to multilingual environments, because the CDA narrative block headings can be autogenerated to use the

---

[3] The distinction is not uniform implemented, nor clean in the current release, but there is at least a top level hierarchy termed "Record artifact"

relevant local language. If coupled with SNOMED's emerging terminology translation refset capabilities, comprehensive selective or multi-language versions of CDA documents could be generated.

Why not use HL7 Templates ?

HL7 have attempted to allow better standardisation of concepts with the draft template specification. However, this specification is incomplete and, whilst purporting to constrain, an HL7 Template is an amorphous concept with a formal definition of:

"A template is an expression of a set of constraints on the RIM or a RIM derived model that is used to apply additional constraints to a portion of an instance of data which is expressed in terms of some other Static Model. Templates are used to further define and refine these existing models to specify a narrower and more focused scope". [4]

Various parts of a CDA document specification can be constrained by an HL7 Template. In fact, multiple HL7 Templates can constrain the same portion of a CDA document specification. HL7 Templates are expected to have a TemplateID and be stored in some repository. Currently most CDA Template constraints have been implemented in Schematron, and so are limited primarily for validating CDA document instances, rather than as a useful tool for computer processing of specification artefacts that might assist the generation or parsing of CDA documents. There is no mechanism, for instance, for in-memory RIM object generation of, or from, these CDA Templates. Nor does it appear to be a capability expected from an HL7 Template given a Template's representation as:

- *"a formal definition in one or more human readable languages or notations*
- *[optionally] a formal definition as a static model*
- *[optionally] one or more implementation specific representations that can be used to validate instances in a particular context"*

This also means that HL7 Templates are unlikely to be able to assist systems to correlate or derive CDA narrative blocks with/to their corresponding Entries.

Issues

The use of archetypes and corresponding SNOMED identifiers is not without some issues, including:-

- Archetypes are specifications for maximal agreed data that can be recorded about a clinical concept. Rather than have CDA entries conform to *open*EHR Archetypes, a relevant restricted subset of an archetype (e.g via the *open*EHR templating mechanism) would need to be specified for each entry of a given CDA document specification to restrict the data required for a particular document type or setting.
- Despite its acquisition by the International Health Terminology Standards Development Organization (IHTSDO), SNOMED CT is still a restricted terminology, only available to licensed users.
- An HL7 Observation Act is not the ideal container for a CDA Entry. It would be better for the next version of CDA to introduce a more suitable generic Entry class, and at the same time, address some of the other problems outlined earlier in this article.

---

[4] from HL7 V3 Templates Draft Standard for Trial Use - Feb. 2008 .

Summary

1. Simple HL7 CDA document XML fragments appear enticingly easy to implement

2. Specifying and implementing extensively terminology-enabled coded entry based CDA documents is a daunting challenge, primarily due to design problems and complexity of the HL7 RIM used for each CDA Entry.

3. In particular, the HL7 RIM

   (a) has no mechanism to adequately tie the narrative to the entry(ies), let alone to allow transformation from one form to the other;

   (b) results in inconsistent representation of clinical concepts;

   (c) has no consistent mechanism to support specialisation semantics;

   (d) has no associated tooling to facilitate easy CDA document design or specification communication.

4. These shortcomings can be addressed by using *open*EHR archetypes as the basis for specifying each CDA Entry. This can be achieved compliant with the current CDA R2 specification.

5. By allocating SNOMED CT identifiers to each archetype, the ontology of information recording can be linked to the ontology of real world concepts already in SNOMED.

6. Additional advantages in multilingual environments can be achieved through the combined use of Archetyped CDA Entries and SNOMED CT.

7. The above approach is limited by the current proprietary nature of SNOMED CT and issues of quality in the current SNOMED CT release.

Useful links and references

- [DOL2005] Dolin B.H., Alschuler L., Boyer S., Beebe C., Behlen F.M., Biron P.V., Shabo A HL7 Clinical Document Architecture, Release 2, *JAMIA*, Oct 2005 (doi: 10.1197/jamia.M1888)
- HL7 Version 3 Standard. http://www.hl7.org/v3ballot/html/welcome/environment/index.htm
- openEHR Architecture Overview http://www.openehr.org/releases/1.0.1/architecture/overview.pdf
- Kabak Y., Dogac A., Kose I., Akpinar N., Gurel M., Arslan Y., Ozer H., Yurt N., Ozcam A., Kirici S., Yuksel M., Sabur E. The Use of HL7 CDA in the National Health Information System (NHIS) of Turkey. *9th International HL7 Interoperability Conference (IHIC) 2008*, Crete, Greece, October, 2008.