

## Relationship between openEHR Specifications and HL7v3 Standards

- openEHR is predicated on a few key design principles which differ in emphasis from the HL7v3 standards development process:
  - ISO RM/ODP - separation of information, services, enterprise viewpoints; in other words - a service architecture approach (HSSP is bringing this to HL7)
  - two-level modelling - stable reference model + archetypes & templates
  - orthodox object-oriented 'extension' (i.e. additive) semantics in object models
- the HL7v3 specifications are not designed for the same purpose as openEHR:
  - they don't generally address EHR requirements (they are essentially a new approach to doing messaging).
  - the messaging focus (serialisation) leads to technical problems when implementing the specifications within systems. For example, the data types are difficult to implement, and have some design problems. The final chapter of the openEHR data types document gives some details - see [http://www.openehr.org/releases/1.0.1/architecture/rm/data\\_types\\_im.pdf](http://www.openehr.org/releases/1.0.1/architecture/rm/data_types_im.pdf)
  - the RIM has not proved particularly useful or easy to implement within the EHR domain:
    - Ontologically it presents problems (i.e. taken as a model of reality); it also doesn't include all the semantics needed for the EHR, see [http://www.openehr.org/releases/1.0.1/html/architecture/overview/Output/design\\_of\\_ehr.html#1148047](http://www.openehr.org/releases/1.0.1/html/architecture/overview/Output/design_of_ehr.html#1148047) and following sections. This is not to say that openEHR's model is 'right' (there are no 'right' models only more or less 'useful' ones), but openEHR does represent a closer implementation of broad analysis and research of the problem domain (specifically the 'clinical information' part).
    - technically, v3 design is a kind of hybrid of an object-oriented model and (according to the authors) a model of relational projection, where you include all the possible attributes in top-level classes, and 'remove' them logically in lower level classes that are actually projections (i.e. particular selections of the total possible attributes) of the top classes. There is no doubt that there is value in this approach when close to implementation and the required level of semantic expression has been reached ( a similar approach is taken when designing openEHR templates). However, the semantics of inheritance in HL7v3 is 'subtractive', not additive, between the RIM and derived models (at the outset). This has been called 'upside-down' inheritance by a lot of people, and isn't a common approach in software engineering in health or elsewhere to our knowledge.
  - The refinement methodology for creating new schemas (RMIMs) from the RIM uses this projection/deletion idea, which causes problems for object-oriented system design (although has been done in the HL7 Java SIG). The designers of openEHR decided to use more orthodox modelling methods where abstract classes have few, common attributes and lower classes add to that.
  - In any case, each RMIM (i.e. each new message definition) is a new information schema, which is not consistent with the openEHR approach, which uses a single stable reference model, and archetypes and templates to create domain models (see CDA which offers an alternative approach within HL7). All information in an openEHR system is instances of the openEHR RM; in an HL7v3 message bank, it is instances of multiple message schemas.
  - in HL7v3, the RIM has to be the basis of all other models. In openEHR is an extensible

Reference Model (i.e. if workflow is to be modelled then new classes are added); service interface and archetype specifications are separate models.

- openEHR isn't based on messaging as a paradigm, although of course it interoperates with messages and uses some particular HL7 specifications e.g. units (UCUM spec) to ensure this is possible.
- The CDA specification differs from the rest of v3 in that it offers a single schema and is more aligned with CEN 13606 and openEHR at the level of composition. A lot of work was done to maximise this compatibility, led by Bob Dolan, Dipak Kalra and Thomas Beale. However, openEHR provides an object model of these concepts; CDA release 1 was an XML-schema, and CDA release 2 is an RMIM. As an RMIM there is no means of refinement apart from the use of templates, which thus have to take on the role of openEHR archetypes (semantic specification) and templates (use specification). There is also no easy way to provide this as the commitment to XML means that a schema is already present and further constraint requires tooling that is not generally available.
- HL7 didn't have an equivalent to archetypes. It has a 'template' notion being specified currently, but HL7 templates apply only to HL7 artifacts (i.e. RMIMs, CMETs etc) which are not used in openEHR. The need is for a kind of model that could easily be used to:
  - define clinical concepts
  - control data validation
  - generate GUI display elements
  - be the formal basis for queries (using paths)
- the specifications coming from the HL7/OMG HSSP (services) project are more inline with openEHR system thinking, and the intention is to participate in this process
- we are following the ISO data types activity that will hopefully provide a standard that all communities can use.

In summary a lot of good work and thinking has been done in the HL7v3 effort, but much of it is in the form of technical artefacts and a methodology that many people find difficult to use.

A description of use of standards in *openEHR* is here -

<http://www.openehr.org/releases/1.0.1/html/architecture/overview/Output/standards.html#1141279>

## Recent changes in HL7v3 (2007/2008)

A new version of the HL7v3 data types is currently under development, which is more compatible with the *openEHR* data types.