8 Rules for an archetype author

Sam Heard

1. Get the background material

- Screen shots from current systems
- Data collections that use this data for supporting purposes
- Paper forms used to collect the data
- Text books
- Internet information
- Other archetypes already in the repository

2. Be inclusive

- Maximal data sets
  - Interoperability depends on this!
- Multi-professional, multi-disciplinary
- Determine what is common and general
  - This has to be as easy to use as possible
- Determine the scope of data that is rarer and for a more specialised
  - Consider specialisation if there is a substantial subset of the data used in a narrow field

3. Get it in a usable form

- Mind map
- Wiki
- Attachment of key background documents
- Record changes and reason

4. Don't be scared to have a go

- Use version control so you can see the evolution of the archetype
- Don't be too generic so that your constituency cannot see what you are getting at

5. Refactor (later)

- Consider reusable clusters or structures
- Consider alignment with the openEHR reference model
- Consider alignment with SNOMED-CT or other terminology in use
- Commit as late in the development cycle as possible

6. Know your reference model

- There are many general features of the reference model that ensure things are done the same way for all content
  - Commital
    - Who, what, where, when...
  - Participations
  - Information is unavailable
  - Linking data
- The choice of class to archetype will have impact down stream

7. Don't include negatives

- Don't use the same archetype for a concept and its negative
  - Remember that a statement like 'No PH of asthma' is more likely to be recorded when someone is presenting with wheezing – and at the same time as a diagnosis of asthma will be recorded.
- Use exclusions for statements excluding things
  - No hospital admissions for, no past history of, no known allergy to penicillin...

8. Everyone has to live with this

- Root archetypes, not specialisations, are the basis for interoperability
  - Everyone has to use the structure and constraints
  - It needs to provide for current systems
  - It needs to acknowledge that recordings will be made at different levels of detail
- Keep whatever you can optional
- Do not demand coding unless it is internal