



openEHR, XML and Archetypes

Andrew Goodchild
DSTC

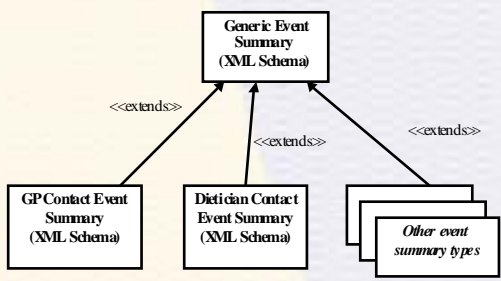
Overview

- XML message standardization strategies
 - Why do archetypes (rather than vanilla XML)
- Archetype validation:
 - how it works
 - examples
 - implementation approaches
- Archetypes compared
 - Archetypes and XML Schema
 - Archetypes and R-MIMs

XML Message Standardization Strategies

XML Message Standardization Strategies

Strategy 1: Every message is independent



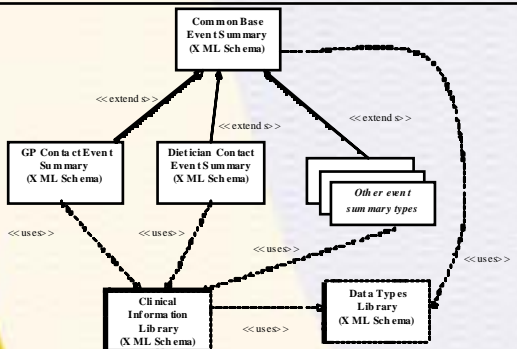
XML Message Standardization Strategies

Strategy 1: Every message is independent

- Advantages
 - Easy to get agreement on a common envelope
 - Simple to implement
- Disadvantages
 - There is no consistency between different message types (e.g. every message type does medications differently or handles capturing the context of observations differently)
 - Special handlers need to be built for every message type (e.g. for querying, viewing, etc)

XML Message Standardization Strategies

Strategy 2: Build a library of clinical models



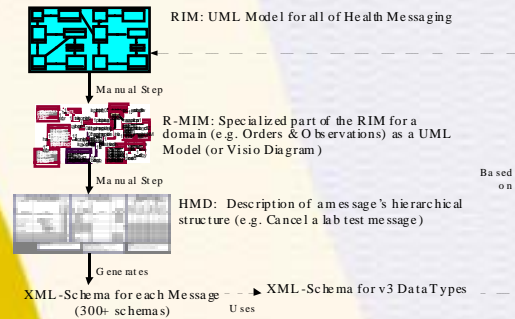
XML Message Standardization Strategies

Strategy 2: Build a library of clinical models

- Advantages
 - More consistency between message types
 - Common Library of clinical concepts
- Disadvantages
 - Moved the problem down a layer: No common structure clinical library components – not model based
 - E.g. A lab result does not necessarily have the similar structure as a full blood count lab result
 - Every time a new type is added to the clinical type library, many messages may need to be fixed
 - The choice of clinical types in messages cannot be generally constrained E.g. “any observation clinical type from HealthConnect can be put in here”

XML Message Standardization Strategies

Strategy 3: Build a methodology (ala V3)



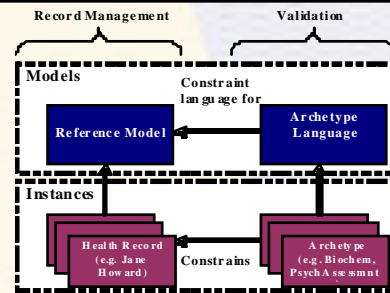
XML Message Standardization Strategies

Strategy 3: Build a methodology (ala V3)

- Advantages
 - Even more consistency between message types (model based)
 - E.g. we can build laboratory observations out of common building blocks, such as ACTS
- Disadvantages
 - The process is extremely time consuming to develop a set of messages
 - Not truly model based: Every message is now totally independent. Very difficult to build one piece of software to handle different messages generically
 - Difficult to deal with changing clinical types

XML Message Standardization Strategies

Strategy 4: The archetype approach



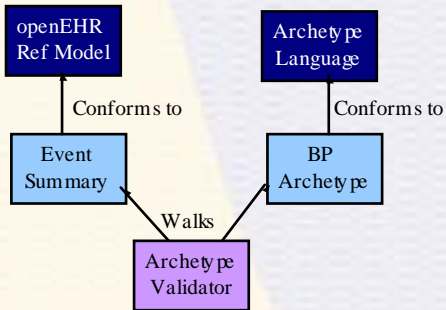
XML Message Standardization Strategies

Strategy 4: The archetype approach

- Advantages
 - Truly Model Based: it is easy to build software that can always view and query data
 - Can deal with change in clinical concepts
- Disadvantages
 - More complex – difficult to explain
 - Need to build a special validation engine for handling archetypes

Arche type Validation

Archetype Validation



Archetype Validation Sample EHR Data

```

<item xsi:type="INSTRUCTION">
  <archetype_id> openEHR.entry.Medication.v1</archetype_id>
  ...
  <action xsi:type="ITEM_LIST" node_id = "NODE0">
    <name> Medication </name>
    <representation node_id = "NODE1">
      <item xsi:type="ELEMENT" node_id = "NODE2">
        <name> Drug Name </name>
        <value xsi:type="DV_CODED_TEXT">
          <value> COUMADIN</value>
          <definition>
            <terminology_id> AZDEX </terminology_id>
            <code_string> 920</code_string>
          </definition>
        </value>
      </item>
    </action>
  </item>
  
```

Archetype Validation Sample Archetype

```

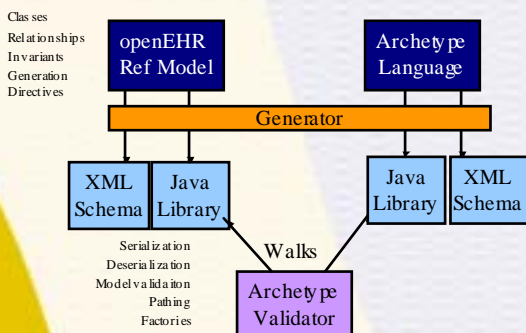
archetype openehr-ehr-en try.Medication.v1
definition --openEHR 0.9
INSTRUCTION [NODE0] matches {
  name matches {
    DV_TEXT matches {
      value matches {[TERM0] -- Medication}}
  }
  action matches {
    ITEM_LIST [NODE1] matches {
      representation matches {
        CLUSTER [NODE2] matches {
          items matches {
            ELEMENT [NODE3] occurrences {1..1} matches {
              name matches {
                DV_TEXT matches {
                  value matches {[TERM1] -- Drug Name}}
                  value matches {
                    DV_CODED_TEXT matches {
                      definition matches {
                        COORDINATED_TERM matches {
                          terminology_id matches {matches {AZDEX}}
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}
  
```

Archetype Validation Sample Archetype

```

terminology
primary_language = en
languages_available = {en,ge}
term_definitions = {
  [TERM0] = {
    language_translation(en) = { text = "Medication"}
    language_translation(ge) = { text = "Medikation"}
    term_binding(SNOMED-CT(2003)) = {24836 4900}
  }
  [TERM1] = {
    language_translation(en) = { text = "Drug Name"}
    language_translation(ge) = { text = "Drogenname"}
    term_binding(SNOMED-CT(2003)) = {36409 0009}
  }
}
  
```

Building an Archetype Validator How DSTC built theirs



Archetypes and other methodologies

Archetypes and XML-Schema Restriction

Archetypes are like a secondary XSI type to validate an element against

```
<action xsi:type="ITEM_LIST" node_id="a9001"> -- list
  <name> <value>Immunisation</value> </name>
  <representation>
    <item xsi:type="ELEMENT" node_id="a9004"> -- item
      <name> <value>Vaccine Code</value> </name>
      <value xsi:type="DV_TEXT">
        <value>COMVAX</value>
      </value>
    </item>
    ...
  </action>
```

Archetypes and XML Schema Restriction

- Currently in schema it is not possible to provide elements with two different types
 - DSTC is discussing the issue with the W3C XML Schema working group
- However, the archetype goes beyond just this:
 - Terms in archetypes are linked back to terminologies and given meaning

Archetypes and R-MIMs

- R-MIMs are a methodological tool for renaming and customizing classes for use in certain contexts
- Archetypes let you use your existing classes and apply additional constraints on using classes in certain contexts
- The difference?
 - Archetypes don't need for a 3 step methodology
 - Archetypes don't result in an XML schema for each message (and a special handler)

Questions
Comments
Discussion