openEHR
Integrating with data source systems

Thomas Beale
Source systems

- Non-EHR systems, e.g. lab systems, imaging systems, departmental databases, research data
- Providing specific patient data items, e.g. path test results
- Generally have own models and terminology
- Usually available in formats like:
  - HL7v2, Edifact, HL7v3 (in use?), Text, CSV
The Challenge of Legacy Data

Problems:

- No / local / incompatible terminology
- No / minimal / incompatible structuring
- Minimal audit trailing

We want to achieve:

- Homogeneous, shared, longitudinal record – of HIGH QUALITY (not a data dump)
- Retain references to original data
- Long term: some source systems migrate to openEHR back-end
General Architecture

- Introduce openEHR EHR nodes
- Input gateways to accept external data, messages
- Main approach for conversion:
  - Build openEHR template for each message
  - Generate XML schema from template definition
  - Populate XML document from source data
  - Transform to standard openEHR XML (known transform)
  - Use integration engine to do XML processing
Template-based Integration

**Inputs**

- HL7 V2 ADT
  Hospital Admission
- HL7 V2 ORU
  Microbiology Result
- HL7 V3 CDA
  Discharge Summary

**Outputs**

- Template Data
  (E.g. Microbiology Laboratory Report)
- openEHR Composition

**EHR Service**

© Ocean Informatics 2007
Template Data Schema

- XML Schema derived from Clinical template
- Automatically generated
- Element names are domain concepts from archetypes/templates
- Includes most archetype/template constraints
- Single transformation into RM schema for all templates
HL7 v2.x message to TDD

Message source system

HL7 v2.x message

Validates

HL7 v2.x Definition

openEHR Standarised Knowledge Environment

Archetypes

Templates

Generate from Tools

Implementation

Standardised XML Environment

Template Data Document

Validates

Template Data Schema (e.g. Patient Admin)

Content-based Transform

© Ocean Informatics 2007
TDD to openEHR

EhrGate WS

openEHR Composition XML (e.g. Patient Administration)

openEHR RM Schema

openEHR Standarised Knowledge Environment

Archetypes

Templates

RM-based Transform

Commit openEHR Compostion

EhrBank repository

Template Data Document (e.g. HL7v2.3 ADT-A01 XML)
Archetype-based Integration

**Inputs**
- (AUS) ORU^R01 Message
- (HL7) ORU^R01 Message
- HL7 CDA R2, ..., other input formats.

**Archetypes:**
- Report
- Microbiology
- Recommendations
- Follow-up

**Template:**
- Microbiology Report

**Key Steps:**
- Auto generate a Template Data Schema
- Transform data source to Template Data Document, Validate
- Apply generic transform openEHR composition, Validate
HL7 v2.x Transformation Process

1. Generate TDS from the Template Designer based on the archetypes and template(s) required to capture the integrated HL7 message content.

2. Convert HL7 v2.x ASCII message to XML.

3. Based on the HL7 v2.x definition and required TDS-formatted output, write the XSLT script to map the HL7 v2.x XML nodes to TDS format.*

4. Invoke HL7 v2.x to TDD transform.*
Application data to TDD

Application Space

**App. Data**

App. Data Schema

<table>
<thead>
<tr>
<th>openEHR</th>
<th>Standarised Knowledge Environment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Archetypes</td>
<td></td>
</tr>
<tr>
<td>Templates</td>
<td></td>
</tr>
</tbody>
</table>

Generate from Tools

Implementation

Standardised XML Environment

**Template Data Document**

| Template Data Schema |

Validates

Validates
TDD to openEHR

EhrGate WS

openEHR Composition XML

Commit openEHR Composition

EhrBank repository

openEHR RM Schema

Archetypes

Templates

openEHR Standardised Knowledge Environment

openEHR Composition

RM-based Transform

Template Data Document
TDD to Exchange Format

- Template Data Document
- Archetype based standard XSL Transform fragments
- openEHR
- CEN 13606
- CDA (CCD)
- CDA R2
- Validate
- Standard Transform for CCD
- openEHR Display

© Ocean Informatics 2007
# Clinical Extracts Compared

<table>
<thead>
<tr>
<th>openEHR R1.0.1</th>
<th>HL7 CDA</th>
<th>CEN 13606-1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EHR Extract</strong></td>
<td><strong>EHR Extract</strong></td>
<td><strong>EHR Extract</strong></td>
</tr>
<tr>
<td>Extract Chapter</td>
<td>Record Target/Participant</td>
<td>Demographic Extract</td>
</tr>
<tr>
<td>Demographics</td>
<td></td>
<td>Access Policy</td>
</tr>
<tr>
<td>EHR Status/Access</td>
<td></td>
<td>Folder</td>
</tr>
<tr>
<td>Folder</td>
<td></td>
<td>Version</td>
</tr>
<tr>
<td>Version</td>
<td>Clinical Document</td>
<td>Composition</td>
</tr>
<tr>
<td>Composition</td>
<td>Section</td>
<td>Section</td>
</tr>
<tr>
<td>Section</td>
<td>Clinical Statement</td>
<td>Entry</td>
</tr>
<tr>
<td>Entry</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
CDA Transformation Process

1. Build generic CDA Level 2 transform.
2. Build Archetype-specific CDA Level 2 transform fragment for each Archetype.
3. Build Archetype-specific CDA Level 3 transform fragment for each Archetype.
4. Import archetype-specific transform fragments into generic CDA Level 2 transform based on document content.
5. Invoke CDA transform.
Issues

- Terminology
  - No term
  - Local / custom term
- Data
  - Missing
  - Data value splitting/merging
  - Type conversion
- Workflow
  - Based on message type