openEHR: a technical overview

23-02-2021
## Agenda and Speakers

<table>
<thead>
<tr>
<th>Time</th>
<th>Speaker</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>1005</td>
<td>John Meredith</td>
<td>Welcome</td>
</tr>
<tr>
<td>1015</td>
<td>Ian McNicoll</td>
<td>The openEHR Technical Platform</td>
</tr>
<tr>
<td>1035</td>
<td>Martin Lewin</td>
<td>Getting started with OpenEHR - a developer's perspective</td>
</tr>
<tr>
<td>1055</td>
<td></td>
<td><strong>BREAK</strong></td>
</tr>
<tr>
<td>1105</td>
<td>Stuart Mackintosh</td>
<td>openEHR - What the developer doesn't need to know</td>
</tr>
<tr>
<td>1125</td>
<td>dude, Where's My Data</td>
<td></td>
</tr>
<tr>
<td>1125</td>
<td>Live Q&amp;A Panel with speakers</td>
<td></td>
</tr>
</tbody>
</table>
The openEHR Technical Platform

Ian McNicoll
Director, Fresh EHR, OpenEHR Foundation
openEHR Technical Platform
(by a clinical hacker!!)

Ian McNicoll
Patient-centric—neutral Platform

Apps

Vendor-neutral Information model

Technology-neutral datastore (CDR)
Archetypes and templates

Template underpinning application

Template underpinning application(s)

Archetypes used in template

ISSUE: Tingling feet, feeling tired
WEIGHT: 76kg
Blood Pressure: 124/92
HbA1c: 7.9%
Assessment: Excellent control

ISSUE: High Blood Pressure
WEIGHT: 66kg
Blood Pressure: 192/114 mmHg
Pulse Pressure: 78 mmHg
SpO2: 92%
Assessment: NAD, see 4/92
openEHR overview

https://www.openehr.org/about/what_is_openehr
openEHR Record hierarchy
openEHR data objects

EHR  ehr_id = 5c8a8636-bc98-4441-abd5-e9cf396e8833

COMPOSITION = 'Nursing Observations'

SECTION Vital signs
ENTRY Blood P.
- Systolic: 134 mmHg
- Diastolic: 86 mmHg
ENTRY Pulse
- Rate: 134 mmHg
- Rhythm: Irregular

SECTION = 'Other obs'
General appearance
- Skin colour: Jaundiced
- Behaviour: Normal
- Hydration
- Skin turgor: Reduced
- Capillary Return: 30s

Hydration
Working with an openEHR CDR

• Smart ‘no-code’ datastore which natively stores, retrieves, queries openEHR data via a standard API

• All data **completely** available

• AQL - Vendor-neutral querying

• ‘No-code’ deployment of new clinical content definitions
What do you need to know?

- Specifications
  - EHR
  - Datatypes
  - AQL
  - REST API, AQL
  - UML
- NOT (usually)
  - Archetype Object model
  - Demographics
  - GDL, Task planning

https://specifications.openehr.org/releases/RM/Release-1.1.0/ehr.html
Archetypes and the RM

- Archetypes are built on top of the RM classes and ‘inherit’ their attributes
  - e.g. An Observation archetype such as Blood pressure inherits the attributes of the RM OBSERVATION class
- Archetypes use the RM datatypes
openEHR REST API(s)

Create a new EHR with an auto-generated identifier.

An EHR STATUS resource needs to be always created and committed in the new EHR. This resource MAY be also supplied by the client as the request body. If not supplied, a default EHR STATUS will be used by the service with following attributes:

- `is_queryable`: true
- `is_modifiable`: true
- `subject`: a PARTY_SELF object

All other required EHR attributes and resources will be automatically created as needed by the EHR creation semantics.

https://specifications.openehr.org/releases/ITS-REST/latest
Templates

- Templates describe the constraints for a specific end-use dataset
- When a composition is committed to the CDR it is validated against the template
- POST:
  - .opt (XML) template
- GET:
  - fetch a template with a given templateId

```xml
<template xmlns="http://schemas.openehr.org/v1">
  <language>
    <terminology_id>
      <value>ISO_639-1</value>
    </terminology_id>
    <code_string>en</code_string>
  </language>
  <description>
    <original_author id="Original Author">Not Specified</original_author>
    <lifecycle_state>Initial</lifecycle_state>
    <other_details id="MetaDataSet:Sample Set">Template metadata sample set</other_details>
  </description>
...
```
Web template json

- Better Care Ehrscape API only
- ehrBase supporting shortly
- also available from Better Archetype Designer
- much easier to read and parse than the xml-based .opt
- wt2doc https://github.com/bjornna/wt2doc
The EHR

- Register a patient with the CDR by creating an EHR object, associating that with a ‘subjectID’ e.g NHS number or MRN and generating a unique ehrId for that patient.

- At each session, retrieve the ehrId for the patient via their subjectID.
Compositions

- Root ‘document’ for clinical data
- All recorded patient data saved inside a Composition
- Carries unique ID
  - UID::serverID::Version_Suffix
  - 5c8a8636-bc98-4441-9d5-e9cf396e8833::ripple_osi.ehrscape.c4h:1
- All changes will create a new version
- Composition serialisation formats
  - CANONICAL/FLAT/STRUCTURED
Queries

- Simple POST that runs a textual AQL statement or stored query and returns a ‘resultSet’
- Logical querying, not physical
- Full access to the whole data tree

```sql
"SELECT
  c/uid/value as compositionId,
  p/data[0][0][0][0]/value/value as problemName,
  p/data[0][0][0][0]/defining_code/terminology_id/value as problemTerminology,
  p/data[0][0][0][0]/defining_code/code_string as problemCode
FROM eehr
  CONTAINS COMPOSITION c
  CONTAINS EVALUATION p (openEHR-EHR-EVALUATION.problem_diagnosis.v1)
WHERE e/ehr_id/value = '{{ehrId}}'
AND c/archetype_details/template_id/value = '{{templateId}}'
```
AQL: Archetype Query Language
The openEHR data tree
openEHR Analytics and integration

openEHR-FHIR INTEROpen Care Connect STU3 Adaptor

This is a proof-of-concept Web service that exposes HL7 FHIR operations (read, search, conform) from a choice of openEHR CDRs (Ehrscape compliant e.g. Marand ThinkEhr or Ethercis) for a small range of INTEROpen Care-Connect profiles, using the HAPI FHIR stack.

https://github.com/AppertaFoundation/openehr-care-connect-adaptor
Getting started with OpenEHR – a developer’s perspective

Martin Lewin
Staircase 13
Developer Roles vs openEHR interaction

openEHR experience

- **Clinical Modellers (not really developers)**
  Understand clinical and technical implications of modelling to write/update archetypes and templates

- **Technical Architect**
  Understand templates and compositions, Observations, Evaluation, Action, and Instructions, Archetype Query Language (AQL), openEHR repository interfaces, authentication, auditing etc.

- **Developers directly interfacing with openEHR repository**
  Understand practicalities of OPT templates, Web templates, Document Formats (JSON / XML), AQL, REST APIs

- **Everybody Else!**
  None of the above. It’s just normal clinical data in a pre-determined hierarchical templated document
Template types – Operational Template (OPT)

Load into repository
View in tools such as Template Designer or EHR Explorer
  • View Model Hierarchy
  • Create / View Compositions
  • Build AQL Queries
Load into software libraries to generate compositions or model class generation
Create a Web Template
Don’t bother opening it to look!
  • Physical Health OPT Example:
  • Height, Weight, BMI, Waist Circumference, Qrisk Score, Common blood test result
  • Approx. 4000 lines of XML

<archetype_id>
  <value>openEHR-EHR-OBSERVATION.blood_pressure.v1</value>
</archetype_id>
<children xsi:type="C_ARCHETYPE_ROOT">
  <rm_type_name>OBSERVATION</rm_type_name>
  <occurrences>
    <lower_included>true</lower_included>
    <upper_included>false</upper_included>
    <lower_unbounded>false</lower_unbounded>
    <upper_unbounded>true</upper_unbounded>
    <lower>0</lower>
  </occurrences>
  <node_id>at0000</node_id>
  <attributes xsi:type="C_SINGLE_ATTRIBUTE">
    <rm_attribute_name>data</rm_attribute_name>
    <existence>
      <lower_included>true</lower_included>
      <upper_included>true</upper_included>
      <lower_unbounded>false</lower_unbounded>
      <upper_unbounded>false</upper_unbounded>
      <lower>1</lower>
      <upper>1</upper>
    </existence>
    <match_negated>false</match_negated>
  </children xsi:type="C_COMPLEX_OBJECT">
  <rm_type_name>HISTORY</rm_type_name>
Template types – Web Template

Compositions are simply hierarchical documents with data in leaf nodes

Web Template:

- Flattens the tree, removes structural nodes
- Displays the leaf node AQL path
- Displays the leaf node data type

```
{
  "id": "systolic",
  "name": "Systolic",
  "localizedName": "Systolic",
  "rmType": "DV_QUANTITY",
  "nodeId": "at0004",
  "min": 0,
  "max": 1,
  "localizedNames": {
    "en": "Systolic"
  },
  "localizedDescriptions": {
    "en": "Peak systemic arterial blood pressure - measured in systolic or contraction phase of the heart cycle."
  },
  "aqlPath": "/content[openEHR-EHR-OBSERVATION.blood_pressure.v1]/data[at0001]/events[at0006]/data[at0003]/items[at0004]/value/magnitude"
}
```
Entry types and Terminologies

Structural Types:
• Observation, Evaluation, Instruction, Action
• Section, Cluster, Complex Object, Attribute

Value Types:
• Quantity: Real, Decimal
• Date/Time, Duration, Period
• Text
• Coded Text

Coding:
• Any coding Scheme
• Snomed or Internal Codes to a template

Web Template Sample
{
   "suffix": "unit",
   "type": "CODED_TEXT",
   "list": [
   {
      "value": "mm[Hg]",
      "label": "mm[Hg]",
      "localizedLabels": {
         "en": "millimeters of mercury"
      }
   }
   ]
}

"termBindings": {
   "SNOMED-CT": {
      "value": "[SNOMED-CT(2003)::163030003]",
      "terminologyId": "SNOMED-CT"
   }
}
The power of AQL

Querying Archetypes

CONTAINS class

You can query ALL available data for an EHR (Patient)
• Regardless of containing composition
• Regardless of who entered the data or if you were previously aware of it being in the repository

Advantages:
• No need to consider database design and indexing
• AQL syntax familiar
• Powerful, can find all occurrences of data model

Disadvantages:
• Long paths
• Unless constrained, will find all occurrences of data model!

```
SELECT
  e/ehr_id/value as ehrld,
  a/uid/value as compositionld,
  a/composer/name as composerName,
  b_a/data[at0001]/events[at0006]/time/value as time,
  b_a/data[at0001]/events[at0006]/data[at0003]/items[at0004]/value/magnitude as systolic,
  b_a/data[at0001]/events[at0006]/data[at0003]/items[at0004]/value/units as systolic_units,
  b_a/data[at0001]/events[at0006]/data[at0003]/items[at0005]/value/magnitude as diastolic,
  b_a/data[at0001]/events[at0006]/data[at0003]/items[at0005]/value/units as diastolic_units
FROM EHR e
CONTAINS COMPOSITION a
CONTAINS OBSERVATION b_a[openEHR-EHR-OBSERVATION.blood_pressure.v1]
WHERE e/ehr_id/value='6f29ba67-fc44-423e-9417-d6bc1196a06e'
ORDER BY b_a/data[at0001]/events[at0006]/time/value desc
OFFSET 0 LIMIT 100

Limit to composition:
CONTAINS COMPOSITION a[MyClinicalObservations.v1]

CONTAINS (
  OBSERVATION a_a[openEHR-EHR-OBSERVATION.body_weight.v2] AND
  OBSERVATION a_b[openEHR-EHR-OBSERVATION.height.v1]) OR
  OBSERVATION a_c[openEHR-EHR-OBSERVATION.body_mass_index.v2])
```
Technology Example - Integration Engine

Application
Prototype tablet application supporting nursing activities on a ward.

Approach
Integration Engine
- Apache Camel
- Dynamic Routing, Processors

Integrations with:
- Active Directory Authentication
- OpenEHR Repository
- Relational Database
- Existing Patient Index (MPI)
- HL7 and FHIR endpoints services for demographics and lab tests.
Technology Example – JSON Manipulation

Application
Apperta Foundation - “Chart My Health”
Physical health monitoring for mental health patients.
Managing and reporting observations - height, weight, blood pressure and a range of blood test results.

Approach
Middleware protects client app from AQL paths
Clinical data all transmitted in flat JSON documents
Templated compositions with mapped parameters
Templated Queries
• Internal AQL templates
• OpenEhr REST API now contains stored queries

App Endpoint JSON document
```json
{
    "ehrTemplateName": "Physical health app self monitoring-v0",
    "parameterMap": "PhysHealthSelf",
    "recordDate": 1552989458000,
    "contents": {
        "physical_health_app_self_monitoring": [{
            "bmi_time": "2021-02-22T09:59",
            "bmi": 25.0,
            "bmi_unit": "kg/m2",
            "waist_time": "2021-02-22T09:59",
            "waist_circumference": 100,
            "waist_circumference_unit": "cm"
        }]
    }
}
```
OpenOutcomes is a product designed to improve the efficiency with which PROMs (Patient Reported Outcome Measures) are recorded.

Key functionality includes:

- Automating the sending of PROMs questionnaires to patients via SMS or email
- Giving patients the ability to submit questionnaires using their mobile device or computer at home
- Charting PROMs for individual patients to see in changes in health over time
- Advanced reporting using aggregated PROMs data
- Pluggable templates for both PROMs and interventions to support a wide variety of clinical specialities

- OpenOutcomes is currently under development and is backed by two NHS trusts within England
- The first deployment within an NHS Trust is expected to begin late Spring
Example – Object Models and Serialisation

**Application Data**
- PROMs Questionnaire results (Observations)
- Care pathway - details procedures and timing of PROMs

**Approaches**

**Care Pathway - Persistent composition**
- Requires manipulation and application of business logic
- Generate classes matching the template which (de)serialize JSON documents.

**Questionnaire Results - Observations at a point in time**
- Many different questionnaires = many compositions
- Simple contents
- Uses JSON document manipulation
In Summary

For most developers in an organisation:
• You may never need to learn the details of openEHR
• Just another REST API with a structured document format
• Does not constrain development language, software stack, or approach to development

Tips:
• Create working examples for your compositions
• Protect as much of the app stack as possible from long AQL paths
• Consider the context of your AQL queries carefully - do you want every available result?
• Build your queries in the available tools and be careful copying them into your code / stored queries
openEHR - What the developer doesn't need to know

Stuart Mackintosh
OpusVL
Background

About me
- Leading developing business data and communication systems with Free / Open Source (FOSS) since mid 90's
- Discovered Open Source enabled me to address the issues, providing no artificial barriers
- Working with the Open Source health & care community for 7 years
- Understand the outcome, and how to measure it, before starting work

About OpusVL
- Team covering full life-cycle (discover, design, build, deploy, support) - bus card image
- Founded to enable organisations to benefit from benefits of FOSS - done professionally
- Delivered electronic observations to SLaM, billing system to NHS Digital, ERP to trusts

https://opusvl.com
Why openEHR?

- Commissioned the DITO project (2018) - obtained Innovate UK funding 2019
- Focus on improving the Open-eObs application - to identify and reduce patient deterioration
- Evaluated all Open Source options
- openEHR was the only option that satisfied technical, clinical and non-functional requirements
- Technical and clinical communities
- EHRbase - most appropriate implementation of openEHR for us
- Standard interface / language between clinician and developer
FHIR / openEHR?

- OpenEHR:
  - Standard for persistence (storage) of clinical data
- FHIR:
  - Standard for communication of clinical data

*Overlap of data, not of purpose.*
What the developer doesn't need to know....

(but should be aware of)
Developing clinical content

- Job of the openEHR clinical teams
- Global network - CKM
- Local / regional specialist configuration
Other applications

- Apps co-exist
- Need no direct knowledge of each other
- Standardised
  - Clinically and technically
- Different versions of the same application
  - De-pressurise upgrades
Other languages

- Doesn't matter your development language preference
- Data is the common element, not the software
- Different approaches, systems, platforms - works together
EHRbase procurement

- It's Open Source!
  - Apache 2.0 licence - 'permissive'
- No SDK / special license to evaluate, experiment, develop or use
  - however many installations
  - any volume of transactions

*You can purchase professional services if you need them...*
Architecture / platform

- EHRbase plays well with containers
  - run it anywhere
  - without prior system knowledge
- Move the app to different platforms, copy it, test it
Database schema

- Automatically installed and deployed
- Manages template changes, indexes and other database internals

*If you decide you do want to explore, it is open...*
Data replication (BC/DR)

- **EHRbase** - Leveraging Postgres functionality
  - real-time replication
  - RPO = zero (up to the last transaction) on unplanned catastrophic failure
  - used at SLaM, NHS Digital
Audit & compliance scanning

- Use code selection policy
  - Set quality rules on import
  - Security, licence, quality

- Implement scanning tools & automation
  - Inventory of software assets

- Automate
  - Use the computers to audit & report

- Focus on development

*It's Open Source - you can know what's inside*
Building a support network

- The community exists already
  - Apperta
  - EHRbase
  - Postgres

- Engage - they are supportive and keen to help
The next developer...

- Building applications with good disciplines
- Version controlled
- Standard off-the-shelf components

*You might be the next developer...*
What the developer won't need to know....

(but should be aware of)
open-eREACT on EHRbase

- How data is defined
  - Point at a template
  - Construct a request, get a response
  - Self-documenting OpenAPI

- Authentication
  - Works with what you have

- External systems
  - FHIR and others - aggregated
OpenEHR inside...

- Modular
- Abstracted
- Packaged concepts
- Unified API

The developer won't need to know openEHR...
Summary

- Specialists focus on their key areas of expertise
  - Developers can focus on developing software
  - Clinicians work on archetypes and templates
  - Operations teams receive predictable software
- No licence barriers
  - Apache 2.0 - permissive
- Upgradable
  - Versioned
  - Robust
- Auditable
  - Know what you have, and what you can do with it
Dude, where’s my data

Mark Hunt
IT Development Manager, Somerset NHS Foundation Trust
About Me and Somerset NHS FT

Mark Hunt, IT Development Manager

- 20+ Years NHS
- Desktop Support, Network Support, Systems Management and Software Development
- Technical lead for both Trust and Regional projects

Somerset NHS Foundation Trust

- OpenEHR journey started with EPMA 2019
- Better OpenEHR CDR, Better Portal
- COVID Response using OpenEHR / Better Forms
- Acute, Community and Mental Health Merger
- Team of 8 supporting solutions development and implementation
Overview

During this presentation we will look at how we get data in and out of an OpenEHR Clinical Data repository and the technical journey that surrounds that. This will not be a technical deep dive into all the APs, but just highlight some high level use cases.

• Understanding the data
• Data sources: OpenEHR, HL7, FHIR, RESTful, MQTT
• Inbound
• Outbound; Views, Reports, BCP

If anyone tells you all your data will arrive as FHIR and only needs to be stored as FHIR, just buy them a coffee; it’s not their fault they got misguided along the way!

*Disclaimer: Any OpenEHR is based on my interaction with a Better OpenEHR Platform.
Understanding the Data

Whether its data coming inbound, being queried for a front end or for some research project your team needs to understand some key things about it.

- What is it?
- Where does it come from?
- Is it trustworthy?
- How will it be used?

Understanding some if not all of these things will help you understand what Archetypes and Templates to use.

**NOTE:** Times change, use cases change and you can not predict the future, but if you have an OpenEHR CDR it’s far easier to adapt.
Inbound Data source to OpenEHR CDR

To get the source data to OpenEHR we need:

- OpenEHR Template created and added to CDR
  - https://ckm.openehr.org/ckm/
  - https://tools.openehr.org/designer

- Ability Transform Source to Template structure

- Ability to send payload to CDR endpoint

Tools in use at Somerset FT:
- Inter Systems Ensemble
- Nextgen Connect (Mirth)
- NodeJS Developments running under PM2
- API client tools Postman and Insomnia
Example:

Patient App sends ‘How Are You’ data and body metrics

- **What is it?**
The app is a proactive tool to directly engage with a patient to help them better manage their overall mental and physical health.

- **Where does it come from?**
The app duh?! It comes from the *interpretation* by the patient and the devices they have.

- **Is it trustworthy?**
Because it is a record of facts from someone who is untrained and maybe using uncalibrated equipment we need to class it as so.

- **How will it be used?**
Data will be used to analyse patient health status over time. It will be used to inform conversations with the patient. Body metrics could be reused in other clinical settings.

**Outcome**
Template needs to contain reusable archetypes for at least the body metrics. There is not a template that specifically aligns in the international or local CKM so following a quick chat with the OpenEHR community we build one based on their advice. As the data is recorded directly by the patient we need to classify it in the wright way.
OpenEHR CDR Output

Once the data is stored it can be used in multiple different ways such as:

• Visualisation of the composition
• Data items can be used for other assessments
• Reports and Dashboards
• Business continuity
• Sending to other systems
Example:

For business continuity key data needs to pushed to BCP machines every 10 mins.

Scope
Nursing staff use 10 core assessments to help manage patient needs and care. In the event of system downtime the output and key items from these assessments should be available to maintain uninterrupted care. Data should be stored by ward and contain the information for all the patients on that ward.

Solution
- Get a list of patients on ward
- Find EHR id for each patient
- Query EHR View for Each Patient
- Combine data for each patient into XLSX file
- Store File on relevant BCP endpoint
Somerset FT Lessons Learned

• Understand your output requirements
  • Per patient (EHR)
  • Per Encounter
  • System Wide

• OpenEHR CDR is just your clinical data

• AQL is just a query language

• Use JavaScript Views and/or external processing

• Users are not always right

• Engagement is key

• Keep it Simple