Integrating FHIR Support for OpenMRS
A Design in Progress

Suranga Nath Kasthurirathne
What we will cover

• HL7 V2 Vs. CDA Vs. FHIR in 2 minutes
• A VERY brief FHIR Introduction
• OpenMRS and FHIR
• Our arguments
• Our use case
• Design
• Deliverables
What we won’t cover

• An introduction to HI data exchange standards
• An introduction to FHIR
• Why FHIR is better (see http://www.hl7.org/implement/standards/fhir/comparison.html)
• Everything and anything FHIR can do
Introduction
FHIR

• **Fast Health Interoperable Resources**
• The latest and the greatest
• Combines the best features of HL7’s Version 2, Version 3 and CDA
• Published as a Draft Standard for Trial use
• Will *(eventually)* become a full normative specification *(in 2016?)*
HL7 V2 Vs. CDA Vs. FHIR

• Practical applications
  – CDA is restricted to clinical settings. V2 and FHIR can be used in other contexts as well.

• Reusability
  – V2, CDA and FHIR are all built around the idea of re-usable segments, but only FHIR segments maintain truly independent identities

• Human readability
  – V2 offers NTE segments, FHIR and CDA require human readable content for all resources

• Messaging paradigms
  – V2 supports event based messaging. CDA does documents. FHIR does both, plus REST and service models
HL7 V2 Vs. CDA Vs. FHIR Contd.

• Extensibility
  – V2 offers Z segments whose meaning is opaque unless prior communication by sender. In comparison, the meaning of FHIR extensions are discoverable by resolving the URI that defines each extension

• Cleanliness
  – V2 messages are the most cluttered, CDA less cluttered, FHIR least cluttered

• Relationship to non-HL7 Standards
  – FHIR resources can provide direct implementation of functionality from other standards such as DICOM

• JSON
  – FHIR supports JSON
How to imagine a FHIR resource?

- Roughly, a FHIR Resource = V2 Segment = CDA Section
- Is ‘self-aware’
- Can be independently manipulated
- Are defined by profiles
- If a resource contains other resources, it will include only a ‘reference’
FHIR Resources Contd.

• Self awareness
  – “Oh wait, you dropped an OBX segment!” Vs. “Look, this segment should be useful to you too, if you want !”

• Transformers!
  – Constructicons: Multiple robots who have their own identity, but can combine into a larger being for a specific task.
OpenMRS and FHIR
## OpenMRS & Standards: Current status

<table>
<thead>
<tr>
<th>Feature</th>
<th>Status</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HL7 V2 Import</strong></td>
<td>Yes</td>
<td>ADTA08 &amp; ORUR01 in OpenMRS core, RGRTA module(ORU, ADT), CHICA module (ADT, ORU, VXR, VXX)</td>
</tr>
<tr>
<td><strong>HL7 V2 Export</strong></td>
<td>Yes</td>
<td>HL7Query module, RGRTA module(ADT, ORU), CHICA module (ORU, VXQ, VXU)</td>
</tr>
<tr>
<td><strong>CCD Export</strong></td>
<td>Yes</td>
<td>Export CCD module (GSOC)</td>
</tr>
<tr>
<td><strong>CCD Import</strong></td>
<td>No</td>
<td>RGCCD module</td>
</tr>
<tr>
<td><strong>CDA Export</strong></td>
<td>Yes...</td>
<td>CDA Generator module (GSOC)</td>
</tr>
<tr>
<td><strong>CDA Import</strong></td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
Why FHIR for OpenMRS?

- Because FHIR is better
- Because FHIR makes more sense in the long run
- FHIR is the heralded proposed solution to be...
- We need to move with the times
- Bangladesh is already using it!

... Consider the implications
Design Considerations

What do we want?

• A minimum implementable unit
• A minimum implementable unit that advises implementers
• A minimum implementable unit that is actually wanted
Acceptable “excuses”

• Too difficult (for phase one)
• The scope is too broad (for phase one)
• Our implementers won’t care (for now)
• Re-use
  – “Look, we can re-use that.. And that, and that and that..”
Merciless chopping

- Ability to **Import** and export data
- Support the interoperability paradigms REST, **Documents, Messages, Services**
- Support **Tags, bundles, binary resources**
- Well.... We can use XDS...
- Lets implement all FHIR resources!
- Enable search for resource by parameters **X,Y,Z** first name, last name, id **etc. etc.**
- Snapshot or other more sophisticated data exchanges
- **RSS and Atomfeed**
Our primary use case

• Allow users to search for, and export a patient’s vital signs in the form of a FHIR resource

• Is a minimum implementable unit
• Is a well used use case
• Has work that we can build upon
FHIR for OpenMRS: A Design
Presenting our design

• Do’s
  – Understand that this is a design in progress
  – Throw stones
  – Ask questions
  – Suggest alternatives

• Don'ts
  – Think that this is 100% correct. Or 90%.. Or 80%...
Sequence diagram

Use Case: Obtain A Patient's Vital Signs

- getPatientsByName("Bob")
- FindByName("Bob")
- patientList
- resourceList
- toFHIR();
- getPatientByld(1)
- vitalSigns
- observationResource
- getVitalSignsByld(1)
- toFHIR();
Things we need to worry about

- Security
- How to search for FHIR Resource objects
- How to model FHIR Resources
- Converting OMRS objects to FHIR Resources
- FHIR Validation
- Converting FHIR to XML / JSON
- Logging
The FHIR Reference implementation

• Why use the reference implementation?
  – Already exists and works
  – Takes care of FHIR specific work
  – Is managed by the experts
High level architecture diagram

- **FHIR module**
  - Restful Interface & security
  - FHIR Reference implementation
    - Validation
    - Convert to xml/json
    - Atomfeed
    - FHIRPatientResource
    - FHIRObsResource

- **OpenMRS FHIR Resources**
  - PatientResource
  - ObsResource

- **OpenMRS Platform**
  - Model Objects
    - Obs
    - Patient
Designing our FHIR Resources

• What resource can be used to model vital signs?
  An Observation resource
From an Observation resource to vital signs

• Questions that we need to answer
  – What content will remain mandatory / optional?
  – What extensions (If any) are required?
  – What value sets will be our questions and answers?

• Represent these Q’s & A’s in the form of a profile that makes our Observation a vital signs resource
Observation Resource to vital signs: Contd.
To be determined...

• What FHIR Resources does our use case require?
  – Patient, Observation, Practitioner...
  – Will we recursively end up implementing everything?

• What value sets can we use / modify?
  – FHIR work groups define their own, somewhat similar to IHE does...
Deliverables

• An OpenMRS module that allows the export of patient vital signs data
• OpenMRS Developers who are comfortable with FHIR
• A re-usable design that can be applied for other resources / use cases
• Our own profiles, and the expertise to build / edit them
• Friends in the FHIR community
Challenges

- Determining value sets
- Keeping an eye on the larger picture
- Building and maintaining profiles
- Trying to Keep the MVP CIEL Dictionary in sync.
- Getting implemented by users
Thanks to...

• The usual suspects:
  – Dr. Paul Biondich
  – Dr. Burke Mamlin
  – Darius Jazayeri
  – Roger Friedman

• New partners in crime:
  – Grahame Grieve
  – Lloyd McKenzie
  – David Hay
Resources

• OpenMRS project page: https://wiki.openmrs.org/display/projects/Building+FHIR+support+for+OpenMRS

• FHIR Documentation: http://www.hl7.org/implement/standards/fhir

• Mailing lists: fhir@lists.hl7.org / dev@openmrs.org

• FHIR Reference implementation:
  http://www.hl7.org/documentcenter/public/standards/FHIR/fhir-0.0.81-Java-0.81.zip

Contacts: surangak@openmrs.org