OpenMRS as the OpenHIE Shared Health Record : A quick introduction for OpenMRS developers

This document was written mainly for the benefit of OpenMRS developers who are unfamiliar with OpenHIE and the SHR. It contains information collected from a number of sources including the OpenHIE wiki and mailing list discussions.

Introduction

The purpose of this document is to discuss potential changes to the OpenMRS API focussed on improving its performance as the Shared Health Record (SHR) envisioned by OpenHIE.

Currently, the OpenHIE project uses OpenMRS as its preferred SHR. In this capacity, OpenMRS has already been implemented as an SHR under the RHEA project launched in Rwanda. However, there are a number of changes that we as OpenHIE would like to see implemented in OpenMRS. We feel that it is best to discuss these changes with the OpenMRS core team in order to evaluate the possibility of a joined approach forward. We would like to assess the OpenMRS community's attitude toward our suggestions, and whether they should be introduced as core OpenMRS features or as OpenHIE specific additional modules.

The discussion will also cover the possibility of enabling OpenMRS to support certain standards, especially HL7 and IHE (Integrating the Healthcare Enterprise) profiles such as the Cross-Enterprise Document Sharing (XDS) integration profile for document exchange. We also feel that some of these suggestions, especially those related to standards and data storage, may also be relevant to the long term goals of OpenMRS.

What is OpenHIE?

OpenHIE is a Health Information Exchange (HIE) which enables the sharing of health data across information systems. It is responsible for normalizing diverse data, and managing their transmission between databases, facilities and regions. To achieve interoperability, the exchange normalizes the context in which health information is created across four dimensions: 1) who received health services, 2) who provided those services, 3) where did they receive the services, 4) and what specific care did they receive. The SHR is but one of the six open source components which the OpenHIE is composed of.

What is the Shared Health Record?

The SHR is a repository containing the normalized version of data created within the HIE. OpenMRS was selected as the SHR as choice after a detailed review process.

Currently, the SHR specific behaviour is added atop OpenMRS via the SHR Adapter module. This module is responsible for intercepting hl7 based requests / responses to / from the SHR, and managing them accordingly. Other OpenHIE related modules, such as the OpenMRS patient search ext. module have also been built to allow users to query this data satisfactorily.

At present, the major workflows of the SHR include saving and retrieving patients and their encounters via ADT and ORUR01 messages.

What limitations we see in the use of OpenMRS as the SHR

As noted by Ryan Crichton (Jembi) in the OpenHIE wiki page, the key changes we would like to see in OpenMRS are as follows,

1) Support the storage of unstructured data

Due to the necessity to manage unstructured data, we are interested in considering alternative methods other than just traditional RBDMS databases for the SHR. To elaborate, while the existing relational database will continue to be used, we feel that an alternative approach is required to manage unstructured document based data. Primarily, we are concerned with the management of Clinical Document Architecture (CDA) documents.

However, note that not all CDA documents qualify as unstructured data, and should therefore be stored in a non-relational database. The CDA document standard can be broken down into three different levels. Of these, level one is completely unstructured, while level three is fully structured. Given this scenario, structured documents such as loose text can easily be decomposed and persisted in a relational database by leveraging the OpenMRS concept dictionary. However, level one documentation such as images are fully unstructured, and must be stored as such. Therefore, being a document does not automatically qualify any resource to be stored outside the RBDMS.

Furthermore, we must also allow the SHR to support the storage / retrieval of CDA documents via either HL7 or the IHE's XDS profile.

- **Initial plans:**

Unstructured data could be stored via the OpenMRS Complex Observations feature (https://wiki.openmrs.org/display/docs/Complex+Obs+Support). The problem with this approach is that complex observations must be tied to encounters, meaning that any unstructured data we wish to associate with a patient must always be done so via an encounter. Does this satisfy our use case?

Furthermore, maintaining structured data in a file system leads to problems related to its meaningful use. Unless metadata for each file is stored separately in an easily searchable database, looking for specific files can be difficult and resource heavy. And also, how can such data stored in the file system benefit from database scaling?

The possibility of supporting polyglot persistence (manage multiple databases, each which is specialized to manage / deal with a specific type of data) is also appealing. In such a scenario, the RBDMS could exist in parallel with a NOSQL database which is better designed to manage unstructured data.

Options for managing document based data include XDS compliant alternatives such as OpenXDS and HIEOS.

*In theory, OpenXDS behaves in a way very similar to the OpenMRS complex observation management process. Each CDA document is split into metadata and the actual document. The document metadata is stored in a relational database while the document itself is passed onto a persistence layer for handling. The default implementation stores the document onto the file system, while the relational database records a pointer to actual location of the file. Implementers can also use their existing infrastructure for storing documents if they would prefer that over the file system. However, to do so they must develop an adapter that specifics their infrastructure for persisting the document.

Under these circumstances, implementing OpenXDS, and writing a custom adapter which allows the unstructured data to be persisted in a NOSQL database may fulfil our use case and also let us benefit from the use of a non-relational DB.

In terms of OpenXDS, we also have an expert (Odysseas Pentakalos) who is working with us on OpenHIE. Would this be a suitable time to build a module which would allow OpenMRS users to leverage the benefits of IHE profile support?

Another question to consider is, given a scenario where polyglot persistence is implemented, at what level will each database be aware of the other? If we are to implement fully functional polyglot persistence, then even a single object may be split into and persisted in different databases (etc. the patient id and encounter id in the relational database, while the complex observation is stored in OpenXDS, etc.). Unfortunately, supporting this can affect performance, plus require changes to the OpenMRS core. It may be much easier and cleaner to implement a model where the each database is unaware of the other, and is queried independently.

2) Ensure that the SHR can be run as a headless service

We are interested in discussing how the OpenMRS service layer (the api module) can be run independently of the web layers (the web and webapp modules). Instead of these layers, we hope to allow users to interface with the SHR using standards based interfaces for XDS + CDA or profiled HL7 V2. For the SHR, our long term plan is to focus on XDS + CDA for future use, while continuing to support HL7 V2 + CDA for legacy users.

However, note that we are not necessarily speaking of physically removing the webapp / web modules from our SHR. Merely ignoring these modules, and preventing users from using them will also be appropriate for our purposes.

3) Add mechanisms for processing, storing and querying unstructured data within OpenMRS

In the event that fully unstructured data must be queried, this will be done via the metadata which is recorded for each resource, and is persisted separately of the actual document. However, this means making two queries for a single action (i.e, query for required document using the RBDMS metadata, and then use those results to collect the appropriate document from the non-relational database).

We prefer that structured documents be persisted in an RBDMS, as it allows the contents of these documents to be queries easily via SQL.

4) Add hooks at various stages of the data life cycle for tasks such as CDS or data validation

There are three possible methods to implement this feature,

1) Database Triggers

2) Hibernate interceptors

3) Spring AOP

Since we will need to support multiple databases, using database triggers for this purpose may not be the best option.

5) Modify the database for performance based on the SHR use case

Improving database performance is closely related to the OpenMRS community’s efforts to scale OpenMRS, and there is potential to work together on these efforts.

The SHR can expect to deal with an exponential amount of information. Since it may be implemented regionally or nationally, the number of requests / responses received can be of a scale never seen before. Other factors which affect this decision involve ease of maintenance, cost and efforts required to maintain maximum uptime.

*In addition to the performance bottleneck at database level, what other performance issues must we be concerned of in terms of high volume?

The argument for the adoption of IHE and XDS

The OpenHIE community adopted the IHE profile system with due consideration to the fact that our target implementers were mainly based in middle and low income countries. Other evaluation criteria which influenced our selection were the ease of implementation, size efficiency, maturity of tooling and our ability to influence the standard. The OpenHIE community has already seen the release of the IHE Care Services Discover (CSD) profile which focuses on use cases from low resource settings. For a more detailed review of OpenHIE and its relationship with IHE, please refer to,

https://wiki.ohie.org/display/SUB/Standards+for+the+Shared+Health+Record