MEDINFO 2013
C.U. Lehmann et al. (Eds.)

doi:10.3233/978-1-61499-289-9-1075

An Automated Routine Childhood Immunization Approach using *open*EHR

Zhiming Liu, Nafees Qamar*, Jie Qian

United Nations University, International Institute for Software Technology, Macau, SAR China

Abstract and Objective

This project studies a working but manual immunization system in place in Pakistan, subject to concerns such as poor record-keeping, reaching targeted children and unavailability of latest census. We propose an openEHR-based solution, called Light-weight Electronic Traceable and Updatable System (LETUS), which aims at increasing childhood immunization coverage and traceability. Two key modules of the solution include: (1) a service that collects the data from the immunization workers, computes population estimates for particular regions, and creates alerts if the ratio of vaccinations over population in a region falls outside a certain range (over or under), and (2) several "thin client" modules where workers can enter their collected data and receive feedback about the current coverage in their region. The proposed software system can be integrated into existing regional immunization registry systems, and run on the servers of a local health agency to ensure timely reporting. Within each immunization registry systems, children data is sharable by applying openEHR approach. This solution will gradually replace the current record keeping process by employing smart phone applications and web services.

Keywords: Immunization campaigns, openEHR, Healthcare

Introduction and the Proposed Approach

The rapidly worldwide growing telecommunications and information technology infrastructures allow deploying latest software solutions to overcome bottlenecks in costly immunization campaigns. In our context in Pakistan [1], we discover the following main issues that interfere with an effective management of childhood immunization programs: 1) poor record-keeping, 2) reaching targeted children, 3) unavailability of latest census, and 4) efficiently monitoring of immunization workers. The approach proposed in this project is implemented in order to overcome these issue, as well as to support rapid availability of data for policy making. This low-cost solution LETUS is anticipated to increase childhood immunization coverage and traceability. In this paper we discuss our ongoing project by the use of *open*EHR. *open*EHR has been principally designed to record clinical information and help domain experts, yet, the approach we consider in this paper uses an EHR service that is to be based on an *open*EHR kernel, which encapsulates *open*EHR Reference Model and Archetype Model, to ensure the computability of archetypes and templates. Consequently, any archetype-driven software system is capable of getting data in and out of the *open*EHR format EHR via the virtual client APIs. In almost other situation, existing data sources have to be accounted for the issue of interoperability. Typically, the operational environment of a health department needs to store both non-structured and structured data. To satisfy this requirement, we deploy both file and database servers respectively for one regional immunization registry system. Due to space reasons, we only discuss our system architecture (as in the Figure 1. below).

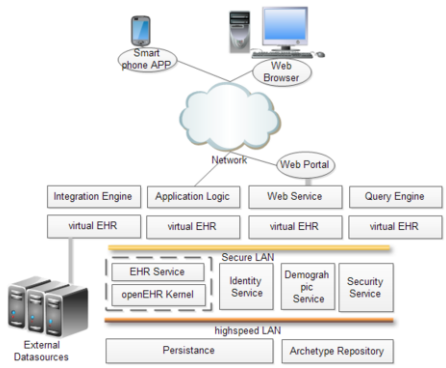

Figure 1.

Architecture Description*:* The overall system architecture of this solution is considered as 5 layers as follows:

1. Persistence: data storage and an archetype repository that holds openEHR designed archetypes, templates and terminologies used to describe the domain concepts in immunization.
2. Back-end services: including EHR, demographic, identity, and security services. The separation of the different services is transparent, and each service has a scalable service interface. In an openEHR approach, EHR service should be based on an openEHR kernel, which encapsulates *open*EHR Reference Model and Archetype Model, to ensure the computability of archetypes and templates.
3. Virtual client: providing APIs to the back-end services. In this layer, the separation of back-end services is hidden, only the functionality is exposed.
4. Application logic: this layer contains specific application services. Particularly, integration engine is to satisfy the requirement of interoperating with existing data sources.
5. Presentation: this layer consists of the graphical interface of the application. This layer can be seen as a "thin-client" and will be installed as an APP on a mobile phone or accessible by a computer connected to the Internet.

Acknowledgments

This work was supported by the project SAFEHR funded by MSTDF. We are grateful to our collaborators in Pakistan from Research & Development Solutions (RESDEV)

References

[1] http://resdev.org/Docs/03immoverview.pdf

Address for correspondence: * *nqamar@iist.unn.edu*