INSTALLING AND SUPPORTING OPEN MEDICAL RECORD SYSTEM (OpenMRS), AN OPEN-SOURCE MEDICAL RECORD SYSTEM, IN AN OPPORTUNISTIC INFECTIONS CLINIC: LESSONS LEARNT AND POSSIBLE EVALUATION STRATEGIES.

By:

Tungamirirai Simbini

A CAPSTONE

Presented to the Department of Medical Informatics and Clinical Epidemiology
and the Oregon Health & Science University
School of Medicine
in partial fulfillment of the requirements for the degree of
Master in Biomedical Informatics (MBI)
May 2008

School of Medicine

Oregon Health & Science University

Certificate of Approval

This is to certify that the Master's Capstone Project of

Tungamirirai Simbini

"INSTALLING AND SUPPORTING OPEN MEDICAL RECORD SYSTEM (OpenMRS), AN OPEN-SOURCE MEDICAL RECORD SYSTEM, IN AN OPPORTUNISTIC INFECTIONS CLINIC: LESSONS LEARNT AND POSSIBLE EVALUATION STRATEGIES."

Has been approved

[redacted]

Mentor/Advisor

May 14, 2008

Table of Contents

Acknowledgements

I would like to take this wonderful opportunity to thank a number of people who have made this work possible.

Mum, Dad, Brother and **Sisters** you have all been asking about this program and here it is.

Benedictor Dube for being the invisible hand in all my work, I want to say thank you. **Ushe** and **Anotidaishe** you came into this world each holding a wonderful gift to give this world, despite adversity you will thrive and realize that gift.

Professor William Hersh for making this program a possibility. You have opened a lot of opportunities to so many in Zimbabwe.

Dr Paul Gorman I am truly grateful for the mentorship through out this program. You have been with me on the ground, experienced our country and advised me most accordingly. Though I walked the path, you definitely showed the way. Thank you.

Vicki and Lloyd Nakashima, for being my new home when I visited Portland, Oregon. You have made this possible in so many ways.

To **Drs Chris Seebregts, David Katzenstein** and **Gerald Kadzirange** for the capacity building at Chitungwiza Opportunistic Infections (OI) Clinic.

Ngonidzashe you made openMRS work despite the challenges we came across in a country undergoing social and political transformation.

And finally the **staff** at Chitungwiza OI Clinic we have all made it happen.

Abstract

OpenMRS is an open source medical record system framework developed specifically for resource-constrained countries battling with HIV/AIDS to scale up the existing information management infrastructure. Chitungwiza Opportunistic Infections clinic in Zimbabwe became the first clinic in the country to install and use OpenMRS. Technical skills to install and support open source technologies are limited the world over and this is worse in resource constrained environments. The transition from paper systems to electronic systems poses challenges to the traditional workflow of health service providers. Electronic systems pose new challenges such as privacy of patient records, integrity of medical record content and maintaining continuity of care. It is important therefore to document these challenges and provide a conceptual framework of integrating electronic systems into day-to-day patient care and evaluating the possible benefits of such systems. Investments into these systems must be evaluated to measure the benefits of such technologies to patient care, public health systems and outcomes research.

Introduction

Background

From October 1980 to May 1981 a case series report was made of five young men, all active homosexuals, treated for an unusual occurrence of *Pneumocystis carinii* pneumonia, suggestive of the possibility of a cellular immune dysfunction predisposing the individuals to opportunistic infections[1]. This disease, later called AIDS, was found to be a result of infection by a retrovirus called Human Immuno-deficiency Virus (HIV). To date close to 40 million people globally are estimated to be infected by this virus[2]. More than 25 million people have died of AIDS since 1981, leaving behind 12 million orphans in Africa alone. Two thirds of all adults and children with HIV globally live in sub-Saharan Africa, with its epicenter in southern Africa. One third of all people with HIV globally live in southern African and 34% of all deaths due to AIDS in 2006 occurred in this part of the world[2].

Patterns of HIV/AIDS in Zimbabwe

The first reported case of AIDS in Zimbabwe was identified in 1985 and by 1990 the first year of national antenatal-clinic based surveillance HIV prevalence was estimated to exceed 10%. By 2003, the country's first national HIV estimate indicated that HIV prevalence amongst adults had reached 24.6%[3]. More recent data from the national antenatal clinic surveillance system suggests the HIV prevalence has begun to fall. Though survey results indicated a fall in Zimbabwe's adult HIV prevalence, these results were taken with caution since trends in antenatal clinic data are subject to bias or changes in the nature or rigor of the surveillance method. To confirm these trends UNAIDS carried out a comprehensive epidemiologic review of data on the HIV epidemic in Zimbabwe between November 2004 and June 2005. Data was collected from all known HIV epidemiological surveillance sources[3]. The evidence available is suggestive of a real decline in national HIV prevalence between 2000 and 2004 a result of a combination of declining HIV incidence and rising adult mortality occurring from the mid- and early 1990s. Sexual behavioral change, in particular a substantial increase in condom use with non-regular partners and an increase in faithfulness, has also contributed to the declines in HIV prevalence and incidence in Zimbabwe.

Scaling up antiretroviral therapy

Despite evidence of declining HIV incidence and prevalence, the estimated HIV/AIDS prevalence in adults (age 15 to 49 years) in Zimbabwe of 20.1% in 2005, with a range from 17.0 to 23.5% is still amongst the highest in the world[4]. There were an estimated 1,391,397 Zimbabweans age 15-49 years living with HIV/AIDS in 2005; 780,000 of whom were women. An estimated 131,370 new HIV infections and 134,990 new AIDS cases occurred among adults age 15 to 49 years in 2005. An estimated 26,610 new HIV infections and 29,467 new AIDS cases occurred among children age 0 to 14 years in 2005. The number of AIDS-related deaths during 2005 was estimated to be 139,950 among adults and 29,150 among children[4].

Less than five years ago all these people would have suffered a steady slow decline in health due to the complete destruction of the immune system and eventually death. The introduction of antiretroviral therapy (ART) has improved quality of life to the infected. ART reduces morbidity and mortality due to HIV and prolongs the lives of those suffering from the disease. In recent years efforts by the World Health Organization (WHO) in collaboration with other UN agencies, non-governmental organizations, national governments and pharmaceutical companies have made it possible for patients in resource limited environments to access this essential medication at very low cost. Today there are concerted international efforts to increase access to ART by patients in resource limited settings.

Supporting information management infrastructure

Successful scaling up of ART depends heavily on an efficient information management system. This information management system must be capable of addressing the 3 core information functionalities in ART namely: patient management, patient monitoring and program monitoring.

Patient management

This is the one to one relationship between a provider and a patient over time assisted by a patient record[5]. In this setting the existing information management systems must be capable of:

- Collecting and storing individual patient data for long periods (stretching into years) as the patient undergoes therapy for life.
- Presenting data to the care provider in a form that is easily readable and must be readily searchable to any given point in the history of care of the patient.
- Assisting the care provider in quickly identifying adverse drug events, possible medical errors, and any complications that might affect the patient's outcome to treatment.
- Giving the care provider an immediate summary of the patient's progress to therapy

Patient monitoring

This is the practice of capturing the history of an individual patient over time and across clinical sites. In this setting the information management system must be capable of:

- Allowing different health care providers within the same clinical setting or across clinical sites access to the complete and accurate history of the patient without loss of data from such factors as illegibility or misplaced parts of the patient record.
- Allowing continuity of care as the patient migrates between clinical sites.
- Tracking and preventing defaulters to therapy
- Allowing effective communication between health care providers and the other supportive clinical and administrative staff.
- Supporting retrospective reporting on referral patterns, formulary compliance, diagnostic test utilization, and preventive care compliance for individual patients.

Program monitoring

This is the routine tracking of priority information about a program and its intended outcomes. In this setting the information management system must be capable of:

- Aggregating patient data and measure the outcomes of the intervention on the population health
- Allowing decision and policy makers make real-time decisions based on real time data aggregates.
- Managing large numbers of patients with minimal input and maintenance costs.
- Improving the quality of patient care through effective knowledge management.
- Supporting secondary data uses such as outcomes research.

Current Information Management Systems

The current paper based information management systems fall short of significantly supporting ART on a large scale. Though paper is easy to use, cheap to acquire and maintain at the point of care it fails to scale up to the voluminous data generated in ART. A patient on ART presents unique challenges to paper based information management systems. In patient management, long-term storage of data becomes cumbersome and voluminous. Health care providers have to spend a lot of time studying historical data in order to get a clear picture of the patient's progress. Patient monitoring across clinical sites is compromised, as patient data is not readily transferable between sites. Patient data can get lost whilst in transit, and data legibility affects the quality of data available to the health care provider and impacts negatively on patient care. Program monitoring relies on expensive follow-up studies depriving the decision makers of real-time results on the progress of the program and its intended outcomes.

Electronic Medical Record Systems

Most of the challenges of the paper-based information management systems can be solved by well-designed electronic medical record systems. Advances in information and communication technologies have made it possible to process huge amounts of data more efficiently. Health care services including ART have realized the potential benefits of utilizing these tools in order to get a well-organized data repository, with real time data access and analysis.

OpenMRS: an open source information management system

For developing countries proprietary health information management systems bring in challenges of affordability. The cost of acquiring such applications is beyond the reach of many public and private sector health services organizations. Efforts by the Regenstrief Institute, Inc - a leader in medical informatics research based in Indianapolis and Partners In Health, a Boston-based philanthropic organization focusing on improving the lives of underprivileged people worldwide through health care service and advocacy; have led to the development of an open source medical record system specifically for HIV/AIDS care and outcomes research. The result to these efforts is the Open Medical Record System (OpenMRS) framework formed in 2004. This system is considered a framework because it is a re-usable design for a medical record software system in which different users can develop and glue the different components and come up with customizable health information management systems (electronic medical record systems) for their own specific uses.

The source code of OpenMRS is available to users to use, change, and improve the software, and to redistribute it in modified or unmodified form hence it is a truly open source application. The aim of open source software is to let the product be more understandable, modifiable, reliable or simply accessible, while it is still marketable. This makes OpenMRS the most ideal information management system in resource-limited settings.

OpenMRS installation at Chitungwiza Opportunistic Infections (OI) Clinic.

Chitungwiza is a dormitory (satellite) town of Harare. It is located about 40km south of Harare. About 1 million people, whose livelihood is dependent on greater Harare, live in the town. The introduction of the opportunistic infections (OI) clinic and ART in Chitungwiza General Hospital has led to the challenges of paper-based systems as highlighted above. The introduction of OpenMRS at the clinic, in mid 2007 was brought about by the failure of paper-based systems to manage patient and program data in HIV/AIDS care. However the introduction of computerized systems poses new challenges to the health care providers who have not had any previous exposure to the technology. Improperly integrated systems can result in unsatisfactory results to the user, leading to lack of interest in the system and hence non-adoption of the technology into the users workflow. In addition, the underlying goal of such systems is to improve the health information management system from the point of care to reporting; hence users must be capable of evaluating the electronic health record system in patient care and program monitoring.

Objectives of the capstone:

1. To describe OpenMRS architecture
2. To document the installation and maintenance process of OpenMRS
3. To describe the lessons learnt from the pilot OpenMRS installation site.
4. To describe the possible integration scenario of OpenMRS in the day to day running of the OI clinic
5. To provide a framework of evaluating electronic health record systems in resource limited settings.

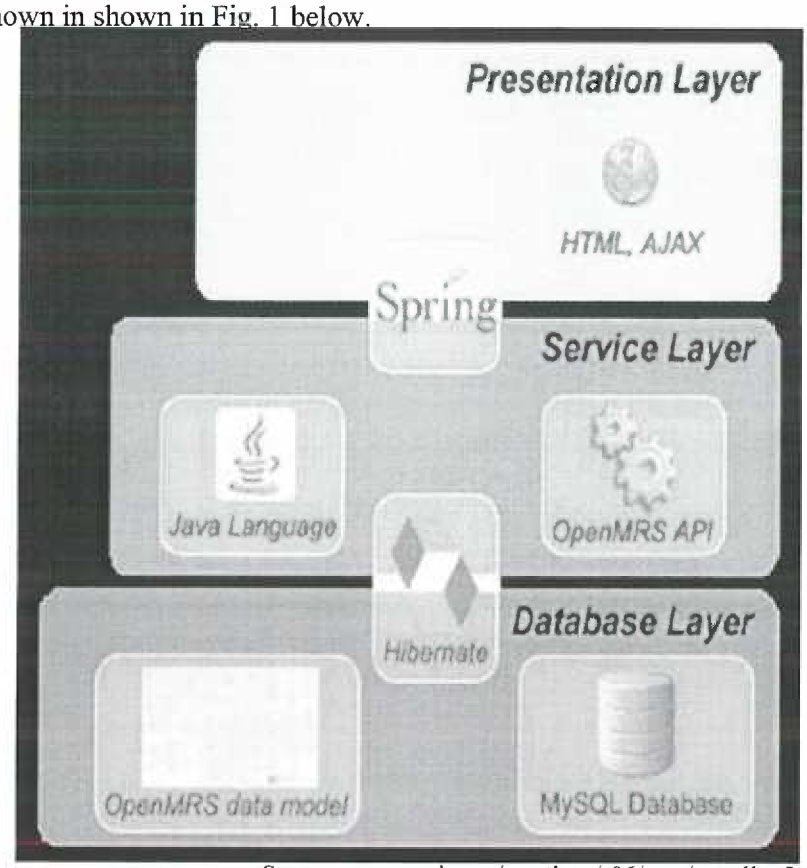
OpenMRS architecture

The design structure of OpenMRS was arrived at based on the challenges of the existing health information management systems in developing countries. The ideal system had to take into consideration the different data being collected at different opportunistic infections clinics, multiple 'silos' of disparate efforts, the non-existence of standards in the health information systems, and had to meet the reporting needs of various stakeholders involved in ART care. Hence the philosophy of OpenMRS was to "enforce deep structure in the database to improve data integrity/quality, scale in size and connectivity, use of informatics standards and open-source based solutions where possible"[6]. With this philosophy in mind OpenMRS was designed as a client-server application i.e. it is designed to work in an environment where many client computers access the same information on a server. It is a Java based application. The application has three code layers namely:

- Database Layer

- Service Layer
- Presentation Layer.

These layers are shown in shown in Fig. 1 below.



Source: www.amia.org/meetings/s06/post/mamlin_5-18_1030a.pdf

Figure 1. OpenMRS Architecture

The database layer

The database layer has two major components, the OpenMRS data model and MySQL database. The OpenMRS data model is heavily borrowed from the Regenstrief model, which is based on a concept dictionary.

The concept dictionary is the fundamental building block of OpenMRS. It defines the names, codes, and other attributes for all medical tests, drugs, and coded results contained in the medical record system. It is the complete reference for the "language" in which medical concepts are represented in a computer. The use of a concept dictionary allows health care providers to capture medical information for multiple purposes. It allows various health care centers to build their own health information systems using the same terminology as other centers but capturing different data variables. This functionality addresses the need of a common interface that can capture different data requirements in different centers using a common language. Just about every medical term that is to be used by the users is captured in the concept dictionary.

To illustrate the significance of the concept dictionary as a building block for the medical information system let us consider the following table:

Encounter Date	**Patient Name**	**Blood Pressure**	**Blood Sugar**	**Urea**
10 May	J Bloggs	120/80	6	1
11 May	A N Other	110/60	8	3
12 May	H Smith	120/70	4	4

Table 1: Data Entry Table

This table records the patient's observations in column format, with each column representing a patient attribute. Each row then represents the complete medical record of the patient. This type of table design becomes difficult to use if one needs to add a new set of patient observations such as liver function tests. In that case one would have to change the structure of the table and add more columns, redesigning the overlying application using the table. With the concept dictionary each observation is not recorded in a column format but there is *one row for each observation* and *not one column* for each observation. Rather than being text, the various observations are actually referenced from a central **concept dictionary**. To add a new observation in this case one has to enter a new concept in the dictionary and it can be referenced. Table 2 illustrates the idea of representing concepts (attributes) per row.

Encounter Date	Patient Name	Observation	Value
10 May	J Bloggs	Blood Pressure	120/80
11 May	A N Other	Blood Sugar	4
12 May	H Smith	Urea	2

Table 2. Recording Patient Observations as references from a concept dictionary

The most "elemental" preferred label for a medical concept within the concept dictionary is a primary term[7]. Each concept has one, and only one primary term. OpenMRS takes into consideration that health information standards such as Systematized Nomenclature of Medicine (SNOMED) or the World Health Organization's International Classification of Diseases (ICD) are not universally used in all areas where the application is implemented. Hence, the definition of a primary term is not based on any of health information standard except that it must as non-ambiguous and as elemental as possible. Primary terms are used to describe the "questions" such as blood pressure value, as well as the "answers" e.g. the patient's blood pressure is 120/80. The meaning of 120/80 is defined in a data type. The data type describes how the stored data looks like such as coded values (terms with coded answers such as Chest X-Rays) or numeric values (terms with numeric results such as serum glucose levels). Each of the primary medical terms is part of well-established medical classes. These classes include: tests (such as laboratory tests), procedures (such as lumbar punctures), drugs, diagnosis (defined medical conclusion) and med set (collection of medications such as (antiretrovirals). A primary concept can be part of or can refer to a bundle of other primary concepts. This bundle of primary concepts is a concept set. An example of a concept set is platelet count is part of a "complete blood count" or CBC (concept set). In medical practice there are different terms that mean the same thing, for example "Full Blood Count (FBC)" has the same meaning as "Complete Blood Count (CBC)". Hence the terms Full Blood Count, FBC, CBC are synonyms of "Complete Blood Count". The concept dictionary defines concept terms, concept classes, concept sets, concept synonyms and concept data types.

Within the database layer is MySQL database, an open source relational database model that forms the backend for data storage in OpenMRS.

The Service Layer

The database layer provides a database application-programming interface (API) that communicates with the service layer. This API is based on Hibernate, a high performance object/relational persistence and query service. Hibernate's primary feature is mapping from Java classes to database tables (and from Java data types to SQL (Structured Query Language) data types). It also provides data query and retrieval facilities. Hibernate generates the SQL calls and relieves the developer from manual result set handling and object conversion, keeping the application portable to all SQL databases, with database portability delivered at very little performance overhead[8]. With this functionality OpenMRS can sit on all SQL based databases. This database API provides a programmatic wrapper around the data model, allowing developers to program against more simplified method calls rather than having to understand the intricacies of the data model [7]. Within the service layer is the Java language in which most of the code development takes place. In this layer is the application API (OpenMRS API). The application API uses collections of API functions to present higher-level functions for developers. This API is based on the Spring framework, an open source application framework for the Java platform. The Spring Framework can be considered as a collection of smaller frameworks or frameworks-in-the-framework. Most of these frameworks are designed to work independently of each other yet provide better functionalities when used together. These frameworks are divided along the building blocks of typical complex applications and they include[9]:

- **Inversion of Control container**: configuration of application components and lifecycle management of Java objects.
- **Data access framework**: working with relational database management systems on the Java platform using Java Database Connectivity (JDBC) and Object-relational mapping tools, such as Hibernate, providing solutions to technical challenges that are reusable in a multitude of Java-based environments.

- **Remote Access framework**: configurative Remote Procedure Call (RPC)-style export and import of Java objects over computer networks supporting Remote Method Invocation (RMI), and Hypertext Transfer Protocol (HTTP)-based protocols including web services.
- **Remote Management framework**: configurative exposure and management of Java objects for local or remote configuration.
- **Testing** framework: support classes for writing unit tests and integration tests.

Presentation Layer

These are the user interfaces. Through the Spring Framework OpenMRS can host multiple user interfaces including web-based front ends or eXtensible Markup Language (XML) based front-ends. To date OpenMRS supports Firefox based web browsers and Microsoft InfoPath forms interface. Microsoft InfoPath is a proprietary application of Microsoft Corporation. It is an application used to develop XML-based data entry forms, and is released as part of the Microsoft Office suite. The main feature of InfoPath is its ability to author and view XML documents with support for custom-defined XML schemas. In InfoPath, the user fills out a form on their computer while it is off-line using the InfoPath client. InfoPath may check some fields on the form for validity, and the user can attach a digital signature. The user later connects to the server and submits the form (in XML form), which may be routed for approval. When the user connects to the server, the form template may be automatically updated[10].

Installation of OpenMRS

From the architecture given above OpenMRS has numerous dependencies. To successfully install OpenMRS a range of expertise are required.

Tomcat expertise:

Tomcat (full name Apache Tomcat) is a web container, or application server developed at the Apache Software Foundation (ASF)[11]. Tomcat provides an environment for Java code to run in cooperation with a web server. It adds tools for configuration and management and includes its own internal Hypertext Transfer Protocol (HTTP) server. The required expertise[7]:

- Install and manage Apache Tomcat
- Upload and install new WAR (short for Web ARchive) files
- Troubleshoot, read log files.

Database Expertise:

OpenMRS uses MySQL as the backend database. MySQL is an open source database that runs on more that 20 platforms including Linux, Windows, OS/X[12] etc. The required expertise[7] are:

- Install and configure MySQL environment
- Understand the OpenMRS data model
- Perform SQL queries and run SQL scripts

Infopath expertise:

The required Microsoft InfoPath expertise[7]:

- Install and manage Microsoft InfoPath
- Advanced Form design
- Understand basic XPath (short for XML Path Language) functions.

XPath is a language for finding information in an XML document. It is used to navigate through elements and attributes in an XML document[13].

Other Expertise:

For successful installation and implementation of OpenMRS there is other expertise required including:

Clinical Form Design

An understanding of how to create meaningful, useful and, non-ambiguous questions/answers. This requires medical expertise, to understand what questions/answers make sense; technical expertise, to understand how questions/answers can be interpreted by a computer and data management expertise to understand how questions/answers will be used for reporting[7].

Dictionary Design

The ability to infer dictionary concepts from a form, modeling expertise such as do you create MYOCARDIAL INFARCTION as a Boolean (true/false)?

Computer Skills

The ability to use the computer is the most basic requirement. Users must be capable of using the mouse for basic navigation, be comfortable navigating through a Windows ® Operating System and Microsoft Office ®Application environments. In addition a supporting team with technical expertise to troubleshoot openMRS and InfoPath applications as well as maintaining the Local Area Network.

Step-By-Step Server Setup.

The step-by step guide on setting up an OpenMRS Windows based server machine can be found at the OpenMRS website: Setting up an OpenMRS server, available from URL: http://openmrs.org/wiki/Setting_up_an_OpenMRS_Server. Additional installation information is also available in the OpenMRS manual available from URL: http://openmrs.org/wiki/OpenMRS_Manual.

Computer Skills for Data Entry Clerks

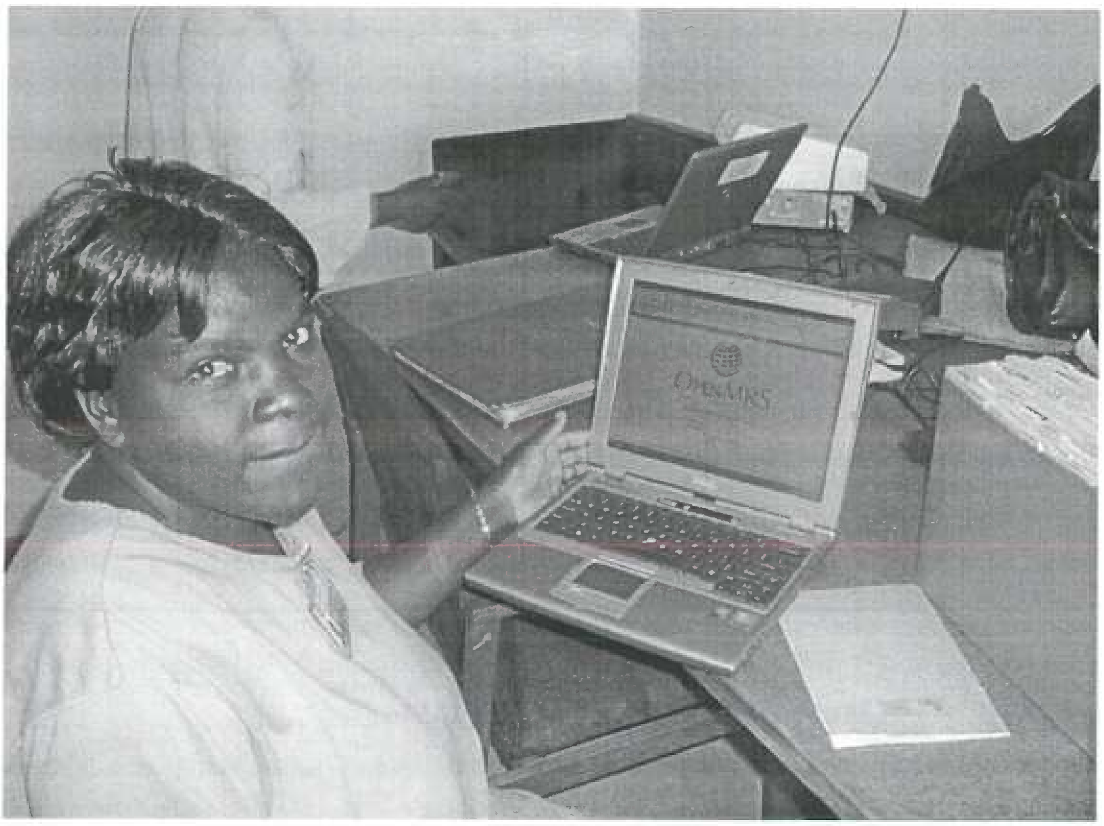
Prior to data entry, 8 data entry clerks were identified from the local community of people living with HIV. These people had no prior experience with computers. The data entry clerks had to undergo a 3 weeks, 6 hours per day computer induction course. Students from the medical school with certified skills from the International Computer Driving License (ICDL) carried out this course. The course covered basic definitions of computer components such as disk drives and computer processing, word and spreadsheet processing. At the end of the course, the data entry clerks practiced data entry using Microsoft Excel spreadsheet to capture four variables of patient data over 3 years As they became more comfortable with data entry and general navigation on a computer, OpenMRS was then introduced.

Setting-Up OpenMRS Local Area Network.

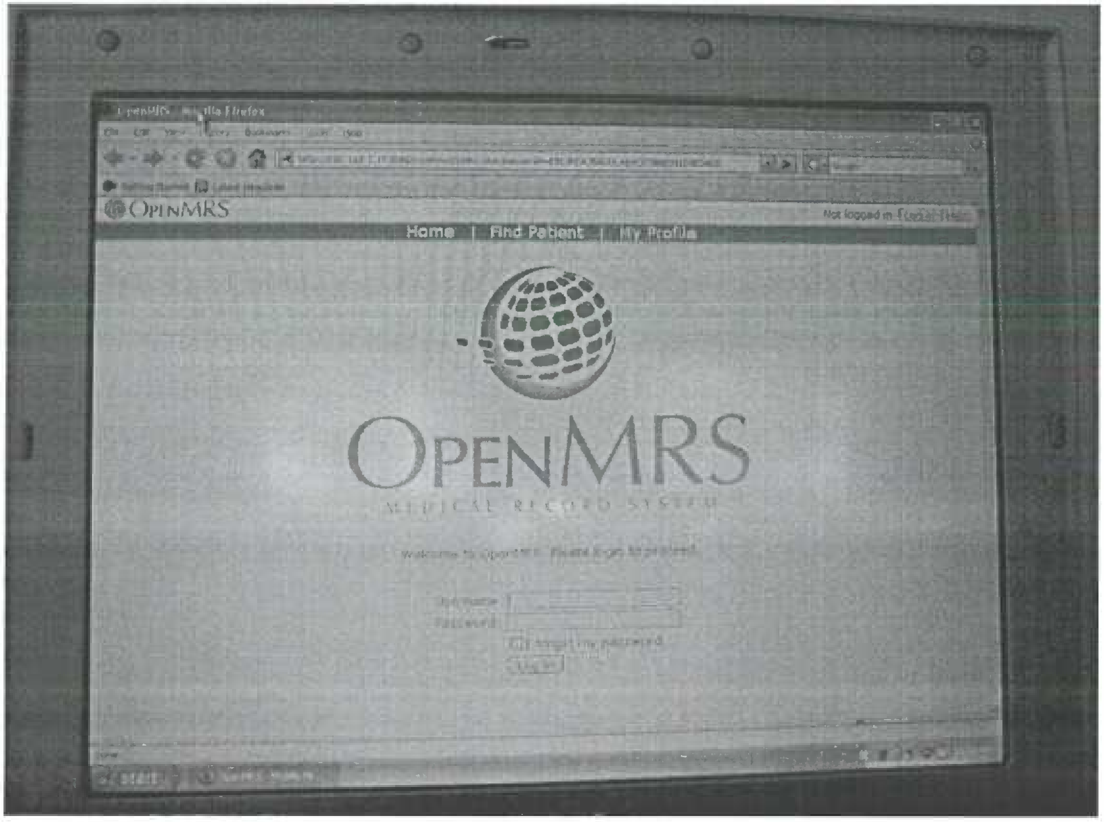
Four laptop computers were networked in a wireless environment for data entry into one central OpenMRS server. Desktop shortcuts pointing to the server URL were added for easier navigation to launching OpenMRS. The site manager was trained on basic network maintenance and troubleshooting to identify and solve simple problems before asking for assistance from the local support office. He was trained on ensuring the server is always switched on for any clients to connect to the server, testing for network connectivity, ensuring each of the clients are pointing at the right server, how to restore the desktop shortcuts if deleted amongst other skills.

For more complex troubleshooting the Center for Evaluation of Public Health Interventions (CEPHI) provided the expertise in supporting OpenMRS. This team depended on the OpenMRS Implementers mailing list for support and Internet access for solutions. Database backups were carried out weekly.

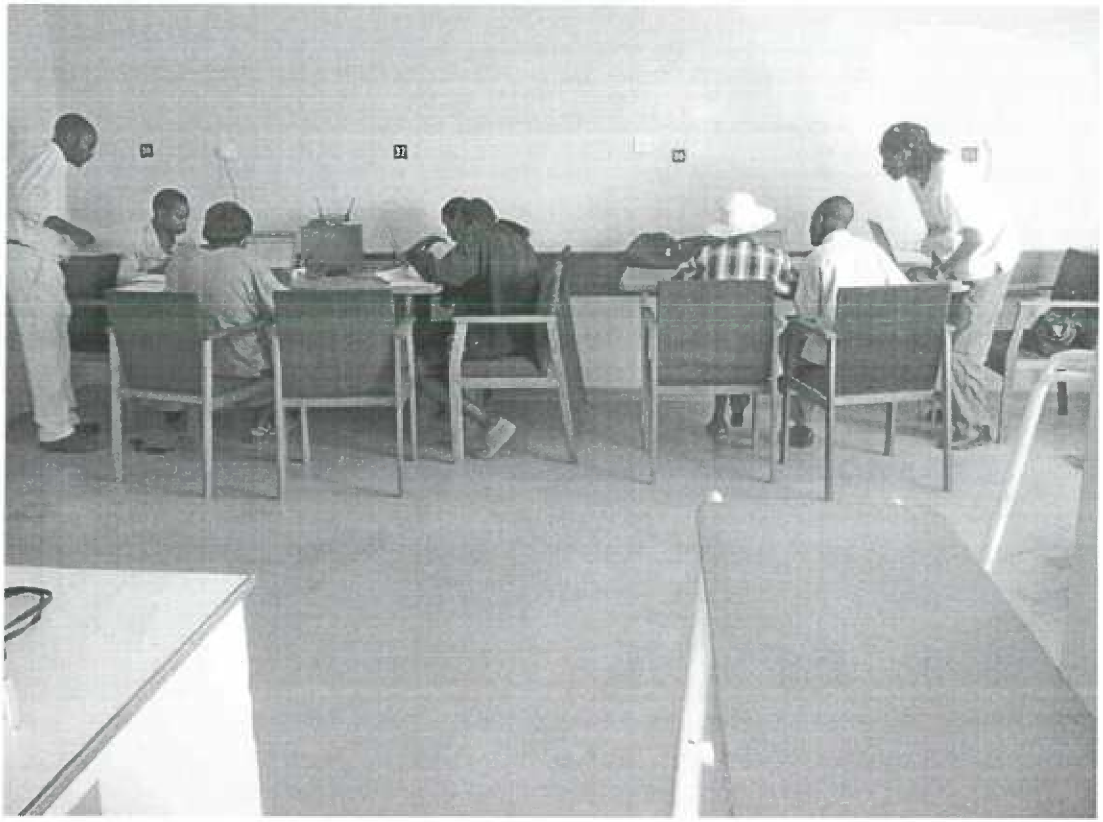
Installation Site in Pictures



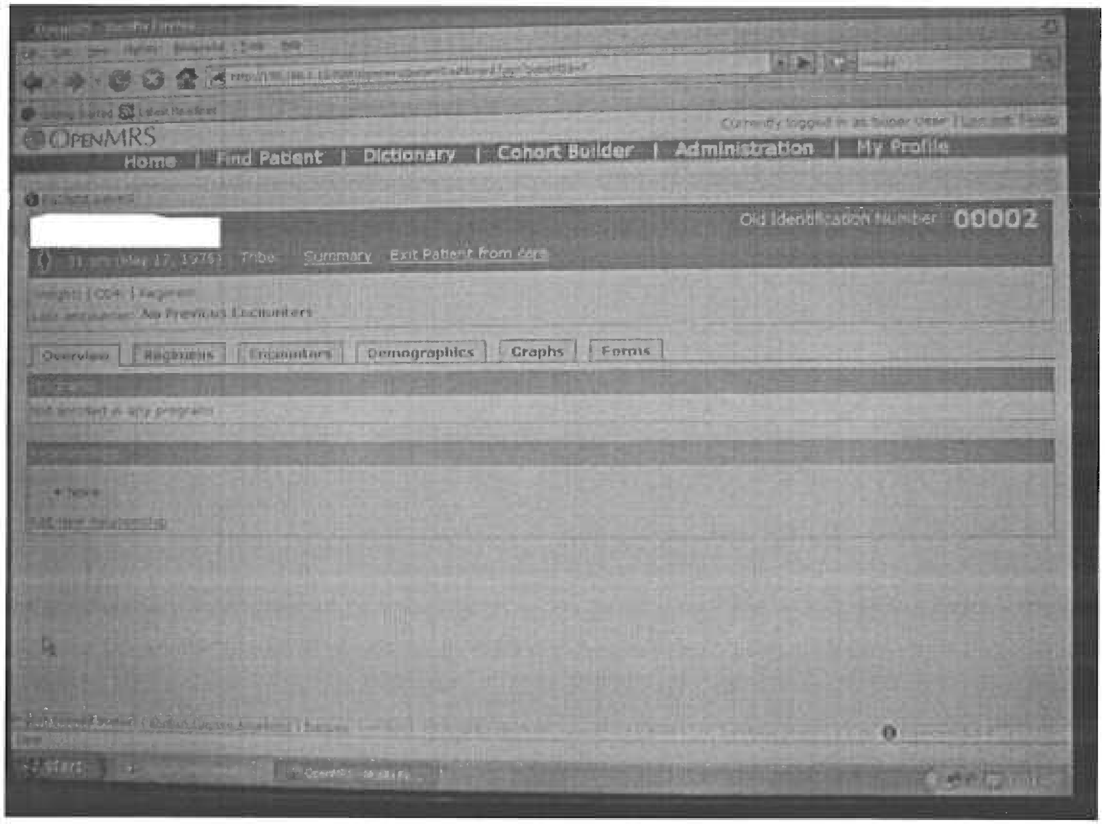
Picture 1: Shows the data entry clerk with OpenMRS running on the laptop. Next to her are the forms she has to enter and behind the laptop is the wireless access point connecting the machines in a local area network.



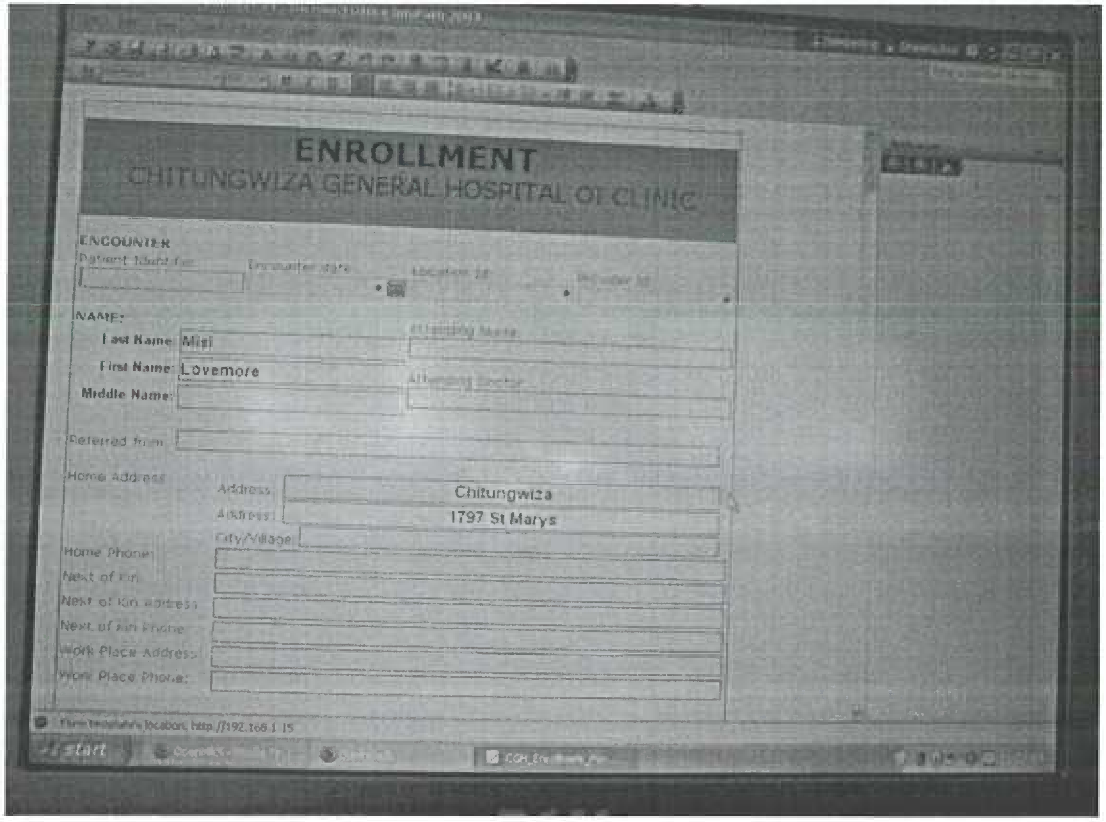
Picture 2: Shows the OpenMRS home page.



Picture 3: The data entry team at work at Chitungwiza OI Clinic.



Picture 4: Typical web-based patient summary screen with the patient name erased.



Picture 5: Data entry screen in Microsoft InfoPath®.

OpenMRS Maintenance and Support

OpenMRS has numerous dependencies. As a result it has various independent points of troubleshooting. Classifying by application, some of the errors encountered in the deployment of OpenMRS are:

Tomcat

Database and WAR file are not compatible. This is the commonest error encountered at the local deployment site. This occurs when there is reinstallation of OpenMRS using a different version of the WAR file. When running OpenMRS the software and database versions are shown in very small text at the bottom of the browser window. The latest WAR file version must have the latest database file. The database version is adjusted by downloading the latest-*mysqldiff.sql* script. Earlier WAR file versions use earlier database scripts.

Other problems include:

- Tomcat not connecting to port 8080. Other programs sometimes use this port. The solution is to confirm no other programs are on this port.
- Tomcat hangs when uploading the WAR file. This typically occurs when a MySQL user account has not been defined.

MySQL

When installing and running MySQL for the first time, one may encounter errors that prevent the MySQL server from starting. The first resource when troubleshooting server issues is the error log. The MySQL server uses the error log to record information relevant to the error that prevents the server from starting. Some of these errors are due to conflicts with other Windows services such as Windows Firewall closing port 3306 or Symantec Antivirus conflicting with MySQL at port 3306.

Microsoft InfoPath

Form design and data capture with Microsoft Infopath can present with its own errors. Sometimes InfoPath does not behave as expected (changes aren't being reflected, etc); clearing the cache often solves the problem. Troubleshooting notes for OpenMRS and Microsoft Infopath can be found at the OpenMRS web site: http://openmrs.org/wiki/InfoPath_Notes.

Results to date

The purpose of the pilot installation was to capture historical data previously captured on paper records from January 2002. These records were piling up over the years and very little of this data was in constant use for patient and program monitoring. OpenMRS was deployed to scale up the information management infrastructure. To this end, the data was meant for longitudinal capture of pharmacy pick-ups, adherence, survival and drug treatment over three years. The data was to be entered from paper records for 2,000 individuals on ART for a median of 1.5 years, 24,000 patient-pharmacy and clinical encounters and 2,000 CD4 laboratory evaluations per year. In addition the pilot was to identify technical, personnel and logistic problems of implementing open-source technologies in resource limited settings.

To date, 213 adult initial patient-pharmacy and clinical encounters have been entered in OpenMRS over a time period of 3 months. This translates to about 70 records per month. The slow data entry process can be attributed to a number of factors. The greatest hurdle to the data entry process was intermittent power supply. About 4 hours per day on average were lost due to persistent power cuts. There is not alternative power supply at the local clinic. Lack of persistent real time technical support to troubleshoot openMRS was another contributing factor. Transport to the implementation site was not always available.

With over 2000 historical records still to be entered very little has changed in terms of the workflow of staff at the clinics. However with this minimum amount of data funding agencies, the Ministry of Health and clinicians have expressed interest in getting to use the data entered in openMRS. Through the Business Intelligence and Reporting Tools (BIRT) reporting module in OpenMRS reports for funders and clinicians can be generated. Through this module data entered in openMRS is validated for errors and to date no errors were reported.

Lessons Learnt

The local pilot site has shown that it is possible to train the local community, without prior exposure to computers, with the appropriate skills and successfully employ their skills locally. Training a local network administrator to troubleshoot some of the more simple errors reduces the costs of traveling to distant sites and also facilitates for remote support through the telephone. The Internet and OpenMRS Implementers mailing lists are essential in order to successfully troubleshoot some of the problems encountered. Ideally, the local site must have some form of Internet connectivity to avoid delays of traveling to a local supporting office. A wireless Local Area Network setup is a cost effective way of setting up a local OpenMRS client-server environment as demonstrated by our local site.

Success Factors to the implementation of OpenMRS

The success of the installation and running of OpenMRS at Chitungwiza OI clinic depended on the use of local resources, collaboration and sharing of resources between the key institutions namely: Ministry of Health, research institutions namely Biomedical Research and Training Institute (BRTI), Medical Research Council of South Africa and the local academic institution namely the University of Zimbabwe. Through pulling of common resources to achieve a common goal it was possible to train the data entry clerks, setup the infrastructure and offer continuous support to the system. Affordable labor was provided from the medical students with previous computer skills training. Laptops were donated from the local research institution. Transport to support the clinic was provided for by the University and the Ministry of Health donated the facility.

Challenges to the implementation of OpenMRS

The major challenge at the installation was intermittent power supply. Electricity to power the machines was not reliable. Efforts to introduce an Uninterruptible Power Supply (UPS) have been considered. However this increases the cost to the installation process. Suggestions have also been made to use solar powered cells to provide electricity but were considered less reliable compared to a UPS.

Accessibility to the site was a major challenge as transport to the clinic was not always available mostly as a result of fuel shortages in the country. The solution was to consider connecting the clinic on the Internet for online support as soon as the power issues have been resolved.

Future Possible Integration Scenarios of OpenMRS into the day to day running of Opportunistic Infections clinics

Electronic data capture, as is done with OpenMRS, can be integrated into the day-to-day running of Opportunistic Infections (OI) clinics in Zimbabwe. The main purpose of electronic medical records systems in ART is to strengthen patient management through such functionality as decision support systems, patient monitoring with functionality such as picking up defaulters and program monitoring for up-to-date reports on patients on ART.

Integrating OpenMRS for Patient Management

Currently OpenMRS is designed to capture the data in its various forms from different ART clinics. To integrate OpenMRS into day-to-day patient management features such as alerting care providers of adverse-drug events, medical errors and complications in patient treatment outcome need must be built in the medical record system. Currently such functionality is not available. Other features such as immediate summary of patient progress have already been built in through OpenMRS's cohort builder.

Integrating OpenMRS for Patient Monitoring

Patient monitoring requires extra functionality of capturing historical data of an individual patient across clinical sites. Health care providers across clinical sites must have access to complete and accurate history of the patient. Such a scenario would require a web-based infrastructure in which patient data is accessed through a secure wide area network.

Tracking defaulters is of paramount importance in OI clinics. OpenMRS can make use of the expanding mobile communication technology in Africa to address this problem. Today's communications technologies are closely aligned to Internet connectivity. Efforts are underway to integrate mobile devices with OpenMRS[14]. The focus of this project is to work on a "transport layer" so that completed forms can be sent over the web, the phone line, or by manually carrying electronic media, and extending the systems ability to present data on the handheld. There is need to expand the electronic health record system interaction with mobile phones. In this case scenario, OpenMRS should be able to pick out defaulters in the system and automatically send them reminders of their review dates. Patients in turn must be able to communicate back with their care providers through OpenMRS using simple text messages.

Framework of evaluating electronic health record systems in resource limited settings.

Electronic health record systems are a fairly new development in developing countries. Their integration into health services must demonstrate some tangible results as the health care providers are working in severely constrained environments already. The addition of a new system that adds on the constraints through its complexity will not be adopted easily. Hence it is important to put in place a monitoring and evaluation framework that is aware of the users' i.e. care providers' needs. In the United States, the rapid growth in electronic health record systems has been driven by the belief that such systems can improve the management of healthcare and in particular can improve quality of care and reduce medical errors[15]. These motivating factors in the developed countries might not necessarily be the same in the developing world. Factors such as accessibility of patient data, use of reminders and reduction of the overall cost of patient care might be the major motivating factors in the developing world. Not enough research is available on the evaluation framework of electronic health information systems. A systematic search of the international literature was conducted in 2003 to identify existing electronic health record evaluation approaches[16]. The principal findings of this study were:

1. No coherent framework for evaluating the impact of electronic health records on health care outcomes and costs was located, although some excellent work had been done in terms of evaluating the impact of subcomponents of the EHR, including the pharmacy and lab/diagnostic test reporting systems.
2. Privacy concerns of providers and consumers are often neglected in the evaluation of health information systems.
3. There is a need for an evaluation framework, which explicitly considers the needs of health care providers, consumers and health system funders.

This paper correctly points out the importance for the evaluation framework to consider the needs of health care providers, consumers and health system funders. The paper rightly concludes by stating that in a single payer (government-funded) health care system such as for most of Africa, the success and impact of health information system projects depends largely on the availability of evaluation information that meets the needs of the key stakeholders involved in the decision to introduce/not introduce these systems on a province by province basis.

Given constrained resources there are key questions that must be answered prior to the future implementation of electronic medical record systems. Some of the key questions that can be asked in this evaluation framework are:

1. Who are the key stakeholders available on site to support the electronic medical record system? This question is important in determining the future continuity, further improvement and ownership of the medical record system.

2. Who are the likely beneficiaries of the system? This is important in determining the overall benefit of the medical record to the health care service providers' community.
3. Are the human, financial and material resources available capable of supporting long-term deployment of electronic medical records? This is important to consider due to high staff turnover to more rewarding career opportunities.

Other key questions that can be asked in the evaluation framework were pointed out by Fraser et al[15]. These include:

1. Does the system improve health care delivery? Some of the measures they suggested for this question would be:
    a. Accuracy, timeliness or completeness of data
    b. Access to data in locations or times that were not previously possible
    c. Communication and/or collaboration
    d. Monitoring or analysis of care
    e. More effective scale up of patient numbers
    f. Reduction of medical errors
    g. Improvement in quality of care – the most important but challenging question
2. What are the downsides of using such a system in a developing country? Possible measures are:
    a. Cost of system development including hardware, networking, software and system configuration
    b. Cost of training, and introducing new software
    c. Ongoing cost of data entry and management
    d. Possible misuse of information

These questions are more relevant when evaluated in the context of the users' expectations and needs. There is need therefore of an evaluation framework that is based on the expectations of the users. Such a framework would involve input from users' feedback from continued exposure to electronic health records and familiarity with the limitations of these systems.

Conclusion

In conclusion, the Chitungwiza OI clinic installation of openMRS has shown that it is possible to use local human resources to install and support open source health care applications. Though open source applications require a steeper learning curve to support and maintain as compared to commercial applications where the commercial vendor supports the users at a cost, training a few dedicated human resources locally can increase the usage of open source applications in the developing countries. Zimbabwe is a country undergoing extreme social, economic and political challenges hence it represented an ideal resource constrained environment to test the sustainability of open source applications. Some of the challenges such as electricity availability and remote site accessibility can be contained by building within the installation model redundancy for alternative solutions.

As user demand increases it is important to consider at the onset an evaluation framework of the value of electronic medial record systems to users, as the resources available are too constrained to leave any room for unnecessary costly investments.

References

[1] Centers for Disease Control. "Pneumocystis Pneumonia-Los Angeles." MMWR. 1981; June 5; 30:250-2.

[2] UNAIDS. Global Summary of the AIDS epidemic December 2006. 2006 [cited 2007 Oct 9]. Available from: URL: http://data.unaids.org/pub/EpiReport/2006/02-Global_Summary_2006_EpiUpdate_eng.pdf.

[3] UNAIDS. Evidence for HIV decline in Zimbabwe: a comprehensive review of the epidemiological data. 2005 [cited 2007 Oct 9]. Available from: URL: http://data.unaids.org/publications/irc-pub06/zimbabwe_epi_report_nov05_en.pdf

[4] Ministry of Health. Health Information and Surveillance Unit. Zimbabwe National HIV/AIDS Estimates, 2005 – A preliminary report. 2004 [cited 2007 Oct 9]. Available from: URL: http://www.unaids.org.zw/unaids_documents/2004Estimatesprelim.pdf

[5] World Health Organization. Interim Patient Monitoring Guidelines for HIV care and ART. 2004 [cited 2007 Oct 9]. Available from: URL: http://www.un.org.vn/who/docs/mekonghivtb/content/29/patientguidelines2.pdf.

[6] Mamlin B, Biondich P. OpenMRS. An EMR for Developing Countries. 2006 [cited 14 January 2008 Available from URL: www.amia.org/meetings/s06/post/mamlin_5-18_1030a.pdf.

[7] OpenMRS Manual. 2006 [cited 14 January 2008]. Available from URL: www.**OpenMRS**.org/images/0/06/**OpenMRS_Manual**_0.1.pdf

[8] Wikipedia. Hibernate (Java). [updated 2008 March 6; cited 2008 March 12]. Available from URL: http://en.wikipedia.org/wiki/Hibernate_(Java)

[9] Wikipedia. Spring Framework. [updated 2008 March 6; cited 2008 March 12]. Available from URL: http://en.wikipedia.org/wiki/Spring_framework

[10] Wikipedia. Microsoft InfoPath. [cited 2008 March 12]. Available from URL: http://en.wikipedia.org/wiki/Microsoft_InfoPath

[11] Wikipedia. Apache Tomcat. [cited 2008 March 12]. Available from URL: http://en.wikipedia.org/wiki/Apache_Tomcat

[12] MySQL. Why MySQL? [cited 2008 March 12]. Available from URL: http://www.mysql.com/why-mysql/

[13] W3Schools. XPath Tutorial. [cited 2008 March 12]. Available from URL: http://www.w3schools.com/xpath/default.asp

[14] OpenMRS. Mobile Data Collection Project. [cited 2008 March 12]. Available from URL: http://openmrs.org/wiki/Mobile_Data_Collection_Project

[15] Fraser SFH, Blaya J, Choi SS, Bonillar C, Jazayeri D. Evaluating the impact and costs of deploying an electronic medical record system to support TB treatment in Peru. [cited 2008 March 12]. Available from URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=1839453.

[16] Neville D, O'Reilly S, MacDonald D, Farrell G, Keough M, Gates K. Measuring the Impact of Electronic Health Records Projects on Health Outcomes and Costs: An Evaluation Framework. March 2004 [cited 2008 March 12]. Available from URL: http://www.nlchi.nf.ca/pdf/final_framework.pdf.