Chapter 40: OpenMRS the Open Medical Record System: Software, Community, Philosophy

Hamish S F Fraser, Ellen Ball, Darius Jazayeri,

Take-home messages:

1. Open source systems have moved from niche products to mainstream tools for medical data management.
2. OpenMRS is an example of an open source EHR system that supports healthcare in Low- and Middle-Income Countries around the world and has fostered large international community developing and implementing the system.
3. OpenMRS supports a wide range of medical coding standards and has helped to foster interoperability and eHealth architecture projects in Rwanda and other countries
4. OpenMRS is unusual for an EHR in that it has a modular architecture which greatly helps with customization and sharing the best ideas and code.
5. A range of customized versions of OpenMRS support care for a wide range of diseases and the system is increasingly used in hospitals.
6. Key challenges moving forward include creating versions of OpenMRS that can be easily configured for use in smaller clinics with limited technical support, and supporting data synchronization between instances of OpenMRS over unreliable networks.

Introduction

High-quality healthcare requires good documentation and communication—therefore, an accurate and complete health record is a core requirement. As discussed in Chapter 10, it is increasingly accepted that each patient should have an Electronic Health Record (EHR). The basic principles of EHRs are covered in that chapter; here we describe an example of an EHR designed especially for health care in Low-and Middle-Income Countries (LMICs) and developed collaboratively with an open approach to software, data standards, and other aspects of design and implementation. We will cover the history and motivation for the project, key milestones along the way, core technical features of the system, and the nature of the collaboration. We will also discuss a number of the challenges encountered in the project, important improvements being made, and the broader contributions of OpenMRS. The OpenMRS website (www.openmrs.org) has extensive documentation and should be consulted along with this chapter.

History and Motivation

In 2000, it was becoming clear that disease burden around the world was increasingly due to chronic diseases and that health care systems needed to change to accommodate that. Much of this burden was due to the increasing prevalence of non-communicable diseases like heart disease, stroke, diabetes, and cancer in LMICs. At the same time, in many low-income countries, especially those in Africa, massive epidemics of two chronic infectious diseases were challenging health systems: HIV and Tuberculosis, including multi-drug resistant TB. As described in Chapter 10, this required a well-organized long term medical record to track the care of patients, including their previous history, treatments, lab results, and clinical course. Important examples of early EHRs were created for several low-income countries particularly to respond to these epidemics, such as Baobab Health in Malawi [Richards 2013]. These systems had to deal with many technical challenges in hardware, software, networking, computer literacy, and training. Many projects sprung up, supported by funding from development agencies, particularly for HIV care: ISante in Haiti [Matheson 2012] and SmartCare in Zambia, DREAM [Nucita 2009] and IQChart [Olivier 2010] in several African countries. Many others were small-scale and specific to particular sites or projects. Each project had to solve most of the same issues and try to rapidly build and support systems with little experience working in these environments. Unfortunately, relatively little sharing of designs or software took place and virtually no evaluations were carried out.

Two projects that started building EHRs at the beginning of the scale-up of the HIV response in 2002/2003 were Partners In Health/Zanmi Lasante working in Haiti (www.pih.org), and AMPATH in Eldoret, Western Kenya, supported by the Regenstrief Institute in Indiana (www.ampath.org). AMPATH deployed an MS Access based system in 2000 for primary care and modified that to support HIV care [Rotich 2003]. PIH adapted their web-based EHR system designed for MDR-TB care for use in Haiti and subsequently Rwanda [Fraser 2004]. Both organizations wished to employ rigorous designs from EHR systems in high-income countries and make those benefits available to health care organizations world-wide, particularly in the poorest and most challenging areas. In 2004, at the Medinfo2004 meeting in San Francisco, the leaders of the two projects met and agreed to create a new EHR system in a very open and collaborative fashion, with a core requirement that the software be open source. The founders were Paul Biondich and Burke Mamlin from the Regenstrief Institute, and Hamish Fraser from Partners In Health. They were soon joined by Christopher Seebregts from the

South African Medical Research Council and Andrew Kanter from the Millennium Villages project.

Work began in late 2004 on an entirely new system developed in Java but based on initial designs and a prototype from AMPATH, and the web-based EHR system in Haiti. At the start of the project a basic design principle was established that the EHR would have three separate layers: the database layer; the middle layer including the user management, security and core business logic; and a user interface layer. This is a standard software architecture but not so common in the EHR world. This would allow the information model to be kept generic and extensible and allow different components to interoperate more easily [Fraser 2005]. The choice was made to code the system in Java to ensure it was flexible to support and extend, and allow the system to run on a range of operating systems.

Deployment History and Scale-up

The first implementation of OpenMRS occurred in Eldoret, Kenya in February 2006, followed by Rwinkwavu, Rwanda in August 2006 and Richmond Hospital in South Africa in September 2006. Within a year, OpenMRS had been deployed in Malawi, Lesotho, Tanzania, and Uganda. Each site had different requirements but used the same core system, which pushed the team to ensure it was flexible and robust. This prompted the decision to create a modular architecture for the system—creating one of the first modular EHR systems worldwide. Also in 2007, OpenMRS was first released under an open source license—a modified Mozilla license. Subsequently, many other organizations started to download and deploy OpenMRS. Notably, the Millennium Villages Project—led by the Earth Institute at Columbia University—adopted OpenMRS for their test villages: Sauri and Dertu, Kenya; Koraro, Ethiopia; Mbola, Tanzania; Ruhiira, Uganda; Mayange, Rwanda; Mwandama, Malawi; Pampaida and Ikaram, Nigeria; Potou, Senegal; Tiby and Toya, Mali and Bonsaaso, Ghana [Kanter 2009]. It is likely that the critical mass of sites, organizations, and countries helped to ensure that the system was stable and highly adaptable.

Many other organizations adopted OpenMRS and by 2011 it was in use in at least 40 countries [Rajput 2011]. Figure 40.1 shows a recent map of known OpenMRS sites; it is likely many other users are not shown. At this point several organizations and countries MOHs had embarked on large scale roll outs of the system. Rwanda was likely the first, starting to roll out the system for HIV care in 2009. There are now over 350 clinics in that country using OpenMRS. A large-scale evaluation study is currently underway funded by the CDC. Kenya commissioned a large-scale rollout in 2012 with support from the USAID and lead by ITech/University of Washington. The system is called the Kenya EMR and is designed to be used at

point of care to support HIV treatment. It can also be updated and managed at a distance [http://www.go2itech.org/where-we-work/kenya]. Bangladesh is now embarking on a large project to rollout OpenMRS. More examples are listed in the section below on “Hospital use of OpenMRS “.

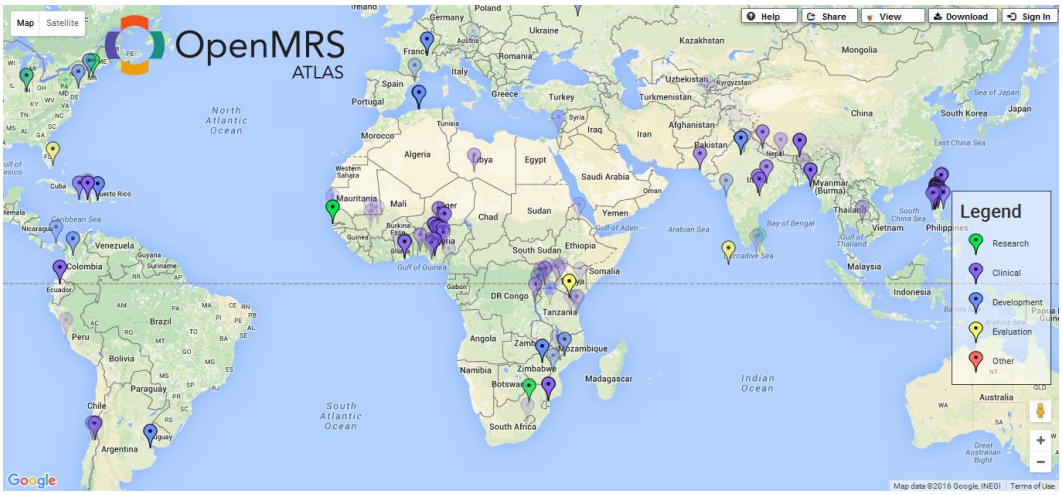

Figure 40.1

An atlas of confirmed OpenMRS sites around the world. See www.openmrs.org/atlas for an up to date version.

Typical Characteristics of an OpenMRS Implementation

Most initial implementations of OpenMRS were to support HIV care and were based in small- or medium-sized clinics. Sites used either Windows or Linux for the operating system. Where possible, sites with internet access were chosen, but many sites had standalone implementations. Power was often unreliable, which led to two potential strategies: (1) collection of data on paper forms with entry by data entry staff and printing of reports, and (2) upgrading the power setup, such as with the addition of solar power or a generator, or the development of a large battery backup. Most sites started with (1) [Mamlin 2006], but now there is a much stronger push to provide reliable systems for direct point of care use. The use of devices with long battery life like laptops for servers, and either laptops or tablet PCs for clients has made this approach much easier. When good internet is available, OpenMRS can be run in a “cloud based” configuration. This was used in the initial Richmond Hospital site in South Africa in 2006 and the PIH sites in Lesotho from 2008, for example.

The Importance of Openness

OpenMRS has a strong philosophy of openness that goes well beyond the source code of the system. The vision of OpenMRS is to support high-quality health care in all communities, even the most impoverished and disadvantaged. This means not only providing "free" copies of the system but giving individuals, health care organizations, and countries ownership of the system to extend and use as they see fit. This openness extends to the "metadata" of the system, such as the concept dictionary and mappings to open standards, and the design and code of reports and decision support tools. A critical innovation is the OpenMRS modular architecture that allows developers to add new functionality to the system without modifying the core code that maintains the stability and security of the system. There are now hundreds of modules which are shared through the OpenMRS module repository. Some of these are designated core modules (such as form entry and reporting tools) and released with new versions of OpenMRS. Others are developed by key partners and may be widely used. Many modules are local enhancements or experiments that do not get beyond the original project. This provides a critical combination of openness and flexibility while avoiding "forking" the core OpenMRS software into multiple versions.

Technical Description

OpenMRS is a flexible, modular, multi-layered system, and one of its strengths is that it can be used in many different configurations. As a result, "OpenMRS" can mean many things. It is built on standard open source components and runs well on Linux or MS Windows [Mamlin 2006].

The minimal version of the system, "OpenMRS Core", is a Java Application Programming Interface (API) and relational data model for storing patient-level clinical and programmatic data, and for managing add-on Modules. This Java API is built on the Spring Application Framework. Data is typically stored in a MySQL database, though some level of database-independence is provided by the Hibernate object-relational mapping system and Liquibase data migrations. The core Java API and data model include metadata that describes the configuration of the medical record, most fundamentally a "concept dictionary", which is used to describe observation data, as well as to provide internationalization and mappings of reference terminologies.

OpenMRS Core includes a legacy user interface as a Spring MVC web application. This user interface is being phased out in upcoming OpenMRS releases, but the majority of currently deployed OpenMRS applications (2015) are likely based on it. OpenMRS Core itself is not sufficient for real-world applications: key pieces of functionality must be provided by add-on modules. For example, most OpenMRS

implementations would include the HTML Form Entry module or the XForms module, both of which let the administrator configure custom data entry screens. The smallest useful configuration of OpenMRS is released as "OpenMRS Platform", and consists of OpenMRS Core and the REST Web Services module; this could be used as back end of a simple medical record system, with some other front end built by a suitably experienced team.

The OpenMRS community releases a "Reference Application" distribution that configures OpenMRS as an EMR for a hospital or small clinic. This distribution includes modules that provide:

- back-end frameworks (UI Framework, App Framework, Reporting, HTML Form Entry)
- common UI elements, shared across different apps (UI Commons)
- common workflows and data representations (EMR API)
- end-user-facing apps (Registration App, Core Apps including a clinical summary, Allergy UI, Admin UI)
- metadata that configures the system with specific forms, diagnosis lists, etc (Reference Metadata)

Other groups have packaged OpenMRS in different ways: Bahmni, a distribution released by ThoughtWorks Inc., uses the OpenMRS web services as the back end for an AngularJS application, which is packaged with an open source Lab Information system (OpenELIS) and an open source ERP system OpenERP/Odoo.

More technical details of OpenMRS and the development process, as well as the source code download links and the module library, are available at www.openmrs.org.

Baobab Health uses the OpenMRS data model for a Ruby on Rails application [Douglas 2010]. It is therefore not strictly a version of OpenMRS, but does allow data compatibility; this allowed the PIH team in Malawi to run both OpenMRS and Baobab off of the same database instance.

Standards and Interoperability

OpenMRS was designed from the beginning to support key informatics standards. The concept dictionary is central to the design of the system. Each data item that can be collected in OpenMRS needs to be added as a concept in the dictionary. Concepts can be Boolean (Yes/No), numeric with ranges, free text, a simple name, or coded. For coded concepts, other simple name concepts represent the options,

such as for a list of possible diagnoses shown in Figure 40.2. Concepts can also be linked in more complex hierarchies (concept sets), for example to group together all tests performed on a blood sample sent to a Hematology lab. The concept dictionary has now been mapped to major medical data coding standards including SNOMED-CT, ICD10, and LOINC. A standard version of the OpenMRS concept dictionary including these mappings is made available by OpenMRS (CEIL Dictionary), and users are encouraged to make use of a subset of that dictionary rather than create new concepts that already exist [Mamlin 2006]. New tools are available to help organizations collaboratively create core concept dictionaries which can help in comparing patient outcomes and sharing of components such as reports and decision support tools—these are part of the "Open Concept Lab" project.

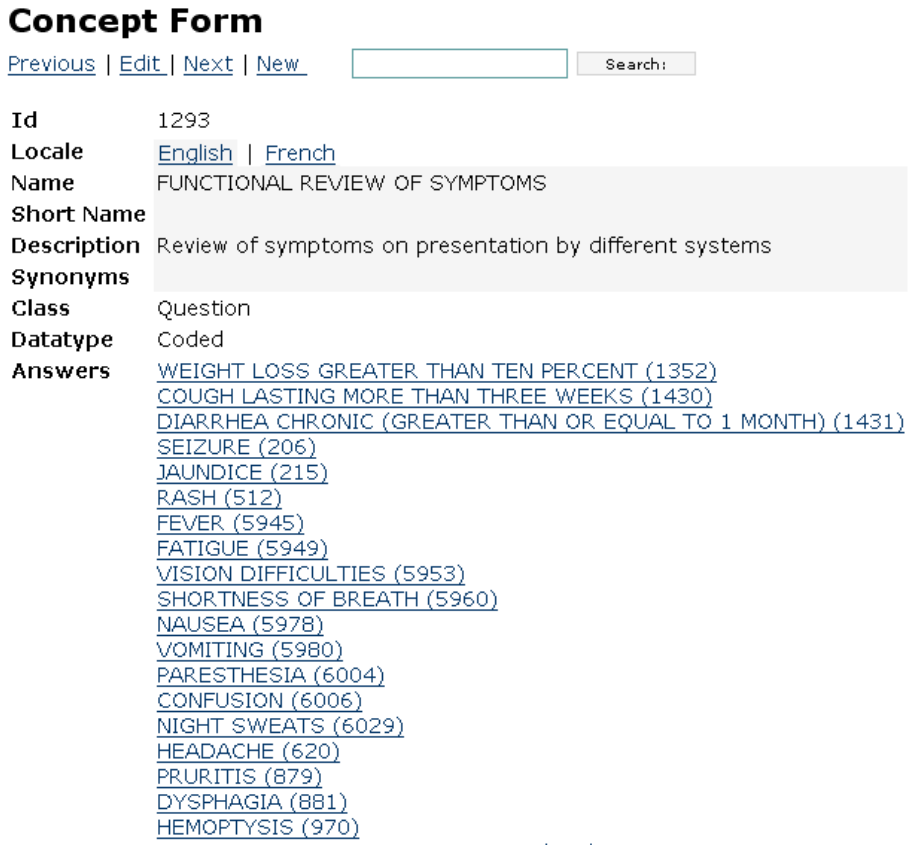

Figure 40.2

A coded Concept showing how simple "name" concepts are linked to the main concept. Numbers are unique Concept IDs

The first OpenMRS implementations included an interface to allow input of HL7 version 2.x messages from completed data forms, and as a method to import data from laboratory information systems. Modules have also been created to allow import of HL7 CDA documents such as Continuity of Care Documents [http://www.hl7.org/implement/standards/product_brief.cfm?product_id=6]. More recently OpenMRS has developed modules to support the HL7 FIHR standard for sharing patient summaries [https://www.hl7.org/fhir/].

Supporting Care for a Range of Diseases

Initial versions of OpenMRS were designed to support HIV care including opportunistic infections like TB. As OpenMRS became more established versions were created adding the capability to support care for a range of diseases. These versions have the same core EHR and were customized by the addition of new concepts, forms and reports. Some add new functionality in OpenMRS modules.

TB and MDR-TB: OpenMRS-TB was developed by PIH to support care of MDR-TB building on the designs and experience in creating EHR systems for MDR-TB care in Peru and the Philippines. It was one of the first versions of OpenMRS to include an integrated set of forms, reports and custom patient summaries and displays [Fraser 2013]. OpenMRS TB was deployed in Haiti by PIH, and in several countries in South and East Asia, including Pakistan, Nepal, and Indonesia. Figure 40.3 shows a screen shot of a summary of a patient's lab results and medication from OpenMRS-TB.

Oncology: An early version of OpenMRS was deployed in Tanzania to support the Ocean Road Cancer Center in Dar es Salaam. In 2010, PIH developed a custom version of OpenMRS to support the new Butaro Hospital Cancer Center in Rwanda in collaboration with the Dana Farber Cancer Center in Boston, USA. This included custom displays and work flows for Oncology and added an order entry system for chemo therapy drugs.

Primary care: A number of healthcare project and countries have deployed OpenMRS to support primary care. In many cases this has included a custom patient registration system using bar code printers and readers, and the issuing of bar coded ID cards to patients. This creates a much more efficient workflow in busy outpatient clinics and is generally popular with patients and staff. PIH deployed such a system in rural Haiti in 2012, based on experience from Baobab Health in Malawi. This approach is often linked to real-time capture of diagnoses and problems [Ball 2012].

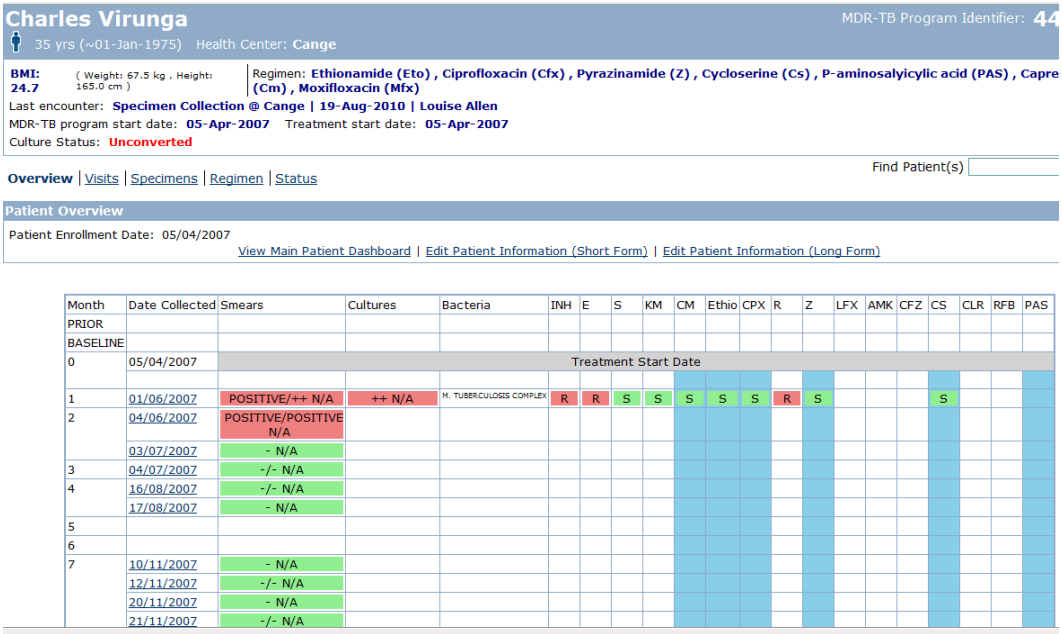

Figure 40.3

Summary and timeline of laboratory and medication data for an MDR-TB patient in OpenMRS-TB.

Heart disease: There are several implementations of OpenMRS to support the management of heart disease. Some PIH/IMB supported MOH sites in Rwanda use a cardiology EMR component to support the care of heart failure, rheumatic fever, and pericarditis [Bukman 2013]. Other organizations including AMPATH have similar add on modules and configuration.

Hospital Use of OpenMRS

A high priority goal for OpenMRS over the last five years has been to build more comprehensive systems to support inpatient hospital care. These types of EHRs need to have good UI and workflow to allow a wide range of staff to register patients and document many different aspects of care. There is typically a larger volume of data than for outpatient systems, and this data can be more complex. Lab data, inventory data, and imaging data, including radiology, is typically important. Several organizations are known to have developed “hospital ready” versions of OpenMRS and there are likely others.

- DHIS Hospital, India: This system was developed over the last half decade to support a group of hospitals in India. It is built on OpenMRS 1.9 and uses several custom modules to provide the functionality and user experience. [http://www.hispindia.org/index.php/products/dhis-hospital]
- Rwanda MOH: The Rwandan MOH embarked on a project to build a version of OpenMRS to support a number of hospitals in Rwanda starting with the Kacyiru Police Hospital in Kigali. This system was developed by a team of programmers employed by the MOH and originally trained in enterprise Java and OpenMRS programming through a course run by PIH in Kigali and sponsored by the Canadian IDRC [Seymour 2010]. This system is being rolled out to ten hospitals in Rwanda at present.
- OpenMRS Reference Application: This project was started in 2010 as an improvement to the existing UI and applications available in OpenMRS 1.9. The initial design work and coding was carried out by a leading OpenMRS architect, Darius Jazayeri, supported by grants from the Rockefeller Foundation. In 2013 the Mirabalais Teaching Hospital (built by PIH), opened in Haiti. In collaboration with ThoughtWorks, Inc., a hospital information system was created based on the initial designs and development work by Jazayeri and the patient registration system developed for the Lacoline Hospital, Haiti [Ball 2012]. The initial system included patient registration, recording of diagnoses and problems, patient summary view for clinicians, a visit view for data clerks, outpatient vs. inpatient statuses, and management of multiple wards and services in a facility. The system is also linked to a commercial radiology PACS system, and has now been extended to cover surgery and other clinical areas. The code from the Mirebalais system was standardized and released as OpenMRS 2.0 in 2014 [http://openmrs.org/2014/02/introducing-openmrs-2-0/]
- Bahmni: ThoughtWorks, Inc. developed a new version of the OpenMRS UI and Application layer in 2013 to use as a hospital EHR for the JSS hospital system in India. This system is also linked to an open source laboratory management system and an Enterprise Resource Management system (see Technical Description above). Bahmni includes new tools for creation of forms and a new drug order entry system based on the OpenMRS order entry business logic. It can support a variety of workflows. It runs on PCs and tablets, and can run in an offline mode on these devices, which lends itself for use in sites with poor infrastructure, [http://www.bahmni.org/].

- The Philippines: The National Telehealth Center has developed a distribution of OpenMRS for health centers with a custom UI. The local governments of Quezon City and Navotas are supporting the implementations of Community Health Information Tracking System "CHITS-Open" built on OpenMRS. The initial system was designed to support primary care, maternal health, and TB care.

Impact to Date

A limited number of studies have attempted to evaluate the impact of OpenMRS on health care processes and even fewer have focused on outcomes. Amoroso showed that OpenMRS could help reduce data errors and improve access to up-to-date CD4 count data [Amoroso 2010]. Were studied the effect of alerts and reminders on actions taken to care for HIV patients in Kenya. He was able to show a fourfold increase in the compliance of staff with treatment guidelines for he targeted items [Were 2013]. Other studies by Were have shown increased ordering of CD4s when prompted by the alerts in patient summaries [Were 2011]. Chapter 24 gives a more in-depth discussion of evaluation techniques and results in for EHRs in LMICs.

Ehealth Architectures and National Rollouts

As the use of EHR systems in LMICs like OpenMRS scales up, there has been increasing focus on interoperability with other eHealth systems. The earliest implementations in Kenya and Rwanda included importation of lab results from laboratory information systems using HL7 2.6. As the role of mHealth applications grew, OpenMRS worked closely with the developers of several open source systems to assist in the import and export of data between them. These systems include CommCare (Dimagi inc), Open Data Kit or ODK (University of Washington), Sana (MIT) and RapidSMS (Columbia University in collaboration with UNICEF). Several mobile versions of OpenMRS now exist that can store data on smart phones or tablets mainly running Android but in some case IOS, and synchronize with OpenMRS.

Another key type of eHealth application in LMICs is used to collect data from health facilities in aggregate form and provide tools to analyze the data at district and national level. The most important example of such a system currently is DHIS2 an open source system developed by HISP and deployed widely by ministries of health in Africa and Asia. It is described in detail in chapter 42. There is an OpenMRS module that allows aggregate data from reports to be automatically uploaded to DHIS2.

Going beyond interactions between two systems is the concept of an eHealth Architecture that links a whole set of eHealth systems. For example, the Rwanda eHealth architecture project links an OpenMRS-based maternal health record to an SMS-based mHealth system tracking community care of pregnant women. The project relies on an interoperability layer to link data between multiple systems. eHealth architectures are described in detail in Chapter 14.

OpenMRS Inc and the OpenMRS Community

OpenMRS started as an informal collaboration between the 4 founders and there organizations and development team. It was then formalized in to a legal entity called an LLC linked to Indiana University with the main function of owning the copyright of the code managing the open source license. In 2011 a new organization was set up as a non-profit corporation in Indiana called OpenMRS Inc. In addition to managing the code, OpenMRS Inc. has a key role in supporting the continued improvement and growth of the software, and in educating developers, implementers, users, and policy makers about the system. OpenMRS Inc. also supports an annual symposium. It was set up with support from the Rockefeller Foundation, and has a five member board, a management team, and an executive director (currently one of the founders Paul Biondich). It has an important role in fundraising and setting the strategic direction of the organization and currently has a small team of programmers and a specialist in online community support.

Also key to the success of OpenMRS is the community that has grown up around the system [Seebregts 2009]. This includes programmers, implementers, users, funders, MOH staff, evaluation experts, private for-profit companies, and others. Development work is carried out by programmers with many different roles, including: working for key partner organizations such as Regenstrief and PIH; other partner organizations like ITech that led the development of the "Kenya EMR"; a wide range of community volunteers; students that are chosen as part of the Google Summer of Code initiative; programmers working for the OpenMRS core team; and partnering companies like ThoughtWorks Inc., IRD, and Jembi. This diversity is an important aspect of how OpenMRS works. Much core work is carried out by salaried, full time programmers that are available to support this code that is "mission critical" for many healthcare organizations. At the same time, the voluntary contributions from individuals and organizations greatly enrich the process, bringing ideas and energy to the work and not infrequently being hired to work full-time on the system. OpenMRS Inc. is currently working with the broader community to increase the input and guidance for the overall project and expand decision making beyond the founding individuals and organizations.

Key Challenges

Training developers: As OpenMRS is built using Java; it requires reasonably high-level programing skills to make significant additions. For sites with limited technical expertise, the goal of OpenMRS is to make the system as configurable as possible and encourage the use of existing distributions and modules. However, there was a strong need for programming skills earlier in the development of the system in countries like Rwanda and Kenya. Innovative training courses were created in both countries to address the skill gaps. In Rwanda, a formal course was set up with support from the IDRC and the Rwandan development board that trained 34 programmers for 9 – 12 months each. The course ran over three years and took computer science graduates mainly from Rwandan universities, who received intensive, hands-on training with experienced programmers led by a PhD computer scientist from the UK, Rowan Seymour. Many of the initial year's class were hired by the Rwandan MOH and worked on the version of OpenMRS for the national rollout for HIV care and the version for hospital use. In Kenya courses were set up for programmers by AMPATH and other organizations. There are now companies in Kenya that can carry out OpenMRS programming.

Increasingly, there are companies that can take on the task of customizing OpenMRS for small or large projects in many low-income countries. Examples include IRD in Pakistan, which has worked in South and East Asia and Africa, and Jembi Inc., a non-profit organization based in Cape Town, South Africa. Another company, ThoughtWorks Inc., is partnering with OpenMRS to develop new modules and UI components of the system and also to assist with implementations including the Bahmni system.

Training users: While good user interface design can reduce the training requirements for users, it is important to ensure that new staff are properly trained. This is a particular challenge in areas where staff have limited IT skills. OpenMRS Inc. provides training materials on the OpenMRS.org site. In addition, partner organizations using OpenMRS provide training courses in the use of the system and in key skills such as data management.

Point of care use and user interface: OpenMRS was initially used mainly in an offline mode with data collected on paper forms and back entered. As more projects had the power and network infrastructure to support point of care use there was increasing focus on the UI and workflow. Initial approaches were specific to one project, such as OpenMRS-TB; now the focus has moved to newer UI frameworks such as OpenMRS 2.0 and Bahmni. A key goal going forward is to encourage the collaborative development and sharing of UI components and business logic in

addition to sharing to "backend" components of OpenMRS. This will require standardizing one or two UI frameworks and programming environments.

Security: OpenMRS has always had a focus on security, utilizing SSL for secure web connections and a role-based authentication system. Regular security audits are carried out. In addition, the key components are open source systems such as MySQL, Hibernate, Apache, Tomcat, and several web development libraries (see technical description above) that have their own support for security updates. As with most systems, the commonest and most serious risks are due to user actions. Training is essential to reducing such risks.

Online and offline synchronization of data: A key challenge of working in low-income settings is the poor infrastructure supporting IT systems. Unreliable power and networking make it very difficult to run systems reliably, whether locally or cloud-based. A good solution is to install a local copy of OpenMRS and synchronize it intermittently to an off-site master server. Technically, this can be complex to program well, but an example was developed by PIH/IMB in Rwanda that has been operational since 2010. Up to 14 smaller clinics synchronize in a bi-directional way to a master server in a larger hospital. This creates a shared record for tracking patients that move through clinics, provides an automatic offline backup, and pushes new lab data such as CD4 counts out to the peripheral sites [Allen 2007, Richards 2013]. Most sites synchronize over the mobile phone data network and use laptop PCs as servers to ensure long battery life and clean shutdown in longer power outages. The system does however need some technical support and monitoring, an updated version is required for general use. An alternative approach is to have a central server connecting "over the cloud" to smaller clinics when the network is available, with clinical staff using tablet PC or smart phone based copies (or simplified versions) of OpenMRS.

Conclusion and Next Steps

Despite the flexibility and scalability of OpenMRS, many organizations need a fairly simple and standard set of functions, at least initially. They also typically lack strong technical support for software development and configuration. Ideally, there would be a small set of "out-of-the-box" OpenMRS packages that could be configured without actual programming requirements to support standard requirements including primary care, HIV care, maternal and child health, etc. These would include a subset of concepts, some forms, reports, and patient summaries. While progress has been made, there are still sites that need additional support to get started with OpenMRS.

The response to the Ebola outbreak is an example of what can be quickly achieved with OpenMRS when there is a strong technical partner. A core team of developers from OpenMRS and ThoughtWorks Inc. created a new version of OpenMRS 2.0 for use in Ebola Treatment Centres (ETC) that included a new user interface for tablet PCs. This was extended to create a new drug order entry system to allow ordering of medications in the infected "Red Zone." The medications were then transmitted to the pharmacy, overcoming the barrier to moving paper-based prescriptions out of the Red Zone. The system also allowed ordering of IV fluids and recording of vital signs and laboratory data. It was built in ten weeks and deployed in Sierra Leone in February 2015 and used to support care for 115 patients. A separate team from Médecins Sans Frontières and Google created a second Ebola EMR also based on OpenMRS and connected to an Android App to allow offline use of the system. They were able to share the same concept dictionary created by the core OpenMRS team demonstrating the benefits of the collaborative and open approach.

Acknowledgements

Dr Hamish Fraser is a Marie Skłodowska-Curie Fellow funded from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 661289 "Global eHealth".

References

Allen C, Jazayeri D, Miranda J, et al. Experience in implementing the OpenMRS medical record system to support HIV treatment in Rwanda. Stud Health Technol Inform. 2007;129(Pt 1):382-6

Amoroso C, Akimana B, Wise B, Fraser HSF. Using Electronic Medical Records for HIV Care in Rural Rwanda. Stud Health Technol Inform. 2010;160:337-41

Ball E, Waters E, Eddy B, Jerome JG, Jean-Paul T, Neumann C, Fraser HS. Clinically Relevant Coded Diagnoses for Developing Countries. Proc. AMIA Symp. 2012

Douglas GP, Gadabu OJ, Joukes S, Mumba S, McKay MV, et al. (2010) Using Touchscreen Electronic, Medical Record Systems to Support and Monitor National Scale-Up of Antiretroviral Therapy in Malawi. PLoS Med 7(8): e1000319. doi:10.1371/journal.pmed.1000319

Fraser HSF, Jazayeri D, Nevil P, Karacaoglu Y, Farmer PE, Lyon E, Smith-Fawzi MK, Leandre F, Choi S, Mukherjee JS. An information system and medical record to support HIV treatment in rural Haiti. BMJ 2004 329;1142-6.

Fraser HSF, Biondich P, Moodley D, Choi S, Mamlin B, Szolovits P. Implementing electronic medical record systems in developing countries. Informatics in Primary Care 2005;13:83–95

Kanter, A.S., Negin, J., Olayo, B., Bukachi, F., Johnson, E. and Sachs, S.E., 2009. Millennium global village-net: Bringing together Millennium Villages throughout sub-Saharan Africa. international journal of medical informatics, 78(12), pp.802-807.

Kwan GF, Bukhman, AK, Miller AC, Ngoga G, Mucumbitsi J, Bavuma C; Dusabeyezu S, Rich M, Mutabazi F, Mutumbira C, Ngiruwera JP, Amoroso C, Ball E, Fraser HS, Hirschhorn LR, Farmer PE, Kamanzi E, Bukhman G. A Simplified Echocardiographic Strategy for Heart Failure Diagnosis and Management at District Hospital Level for sub-Saharan Africa. J Am Coll Cardiol HF 2013;1:230–6

Mamlin BW, Biondich PG, Wolfe BA, Fraser HSF, Jazayeri D, Allen C, Miranda J, Tierney WM. Cooking Up An Open Source EMR For Developing Countries: OpenMRS – A Recipe For Successful Collaboration. Proc AMIA Symp 2006: 529-533

Mohammed-Rajput, N.A., Smith, D.C., Mamlin, B., Biondich, P., Doebbeling, B.N. and Open MRS Collaborative Investigators, 2011. OpenMRS, a global medical records system collaborative: factors influencing successful implementation. In AMIA annual symposium proceedings (Vol. 2011, p. 960). American Medical Informatics Association.

Nucita, A., Bernava, G.M., Bartolo, M., Masi, F.D., Giglio, P., Peroni, M., Pizzimenti, G. and Palombi, L., 2009. A global approach to the management of EMR (electronic medical records) of patients with HIV/AIDS in sub-Saharan Africa: the experience of DREAM software. BMC medical informatics and decision making, 9(1), p.42.

Olivier, B.N., 2010. Rapid model influences population policy orientation for the last 5 years in Rwanda. International Journal of Evidence-Based Healthcare, 8(3), pp.187-188.

Rotich, J.K., Hannan, T.J., Smith, F.E., Bii, J., Odero, W.W., Vu, N., Mamlin, B.W., Mamlin, J.J., Einterz, R.M. and Tierney, W.M., 2003. Installing and

implementing a computer-based patient record system in sub-Saharan Africa: the Mosoriot Medical Record System. Journal of the American Medical Informatics Association, 10(4), pp.295-303.

Seebregts CJ, Mamlin BW, Biondich PG, Fraser HSF, Wolfe BA, Jazayeri DG, Allen C, Miranda J, Baker E, Musinguzi N, Kayiwa D, Fourie C, Lesh N, Kanter A, Yiannoutsos C, Bailey C, The OpenMRS Implementers Network. Int J Med Inform. 2009 Nov;78(11):711-20].

Seymour RP, Tang A, DeRiggi J, Munyaburanga C, et al. Training Software Developers for Electronic Medical Records in Rwanda. Stud Health Technol Inform. 2010;160:585-9

Were MC, Shen C, Tierney WM, Mamlin JJ, et. al. Evaluation of computer-generated reminders to improve CD4 laboratory monitoring in sub-Saharan Africa: a prospective comparative study. J Am Med Inform Assoc. 2011 Mar-Apr;18(2):150-5. doi: 10.1136/jamia.2010.005520.

Were MC, Nyandiko WM, Huang KT, Slaven JE, Shen C, Tierney WM, Vreeman RC. Computer-generated reminders and quality of pediatric HIV care in a resource-limited setting. Pediatrics. 2013 Mar;131(3):e789-96. doi: 10.1542/peds.2012-2072.

