

# AMPATH Medical Record System (AMRS): Collaborating Toward An EMR for Developing Countries

Burke W. Mamlin, M.D. and Paul G. Biondich, M.D., M.S.

Regenstrief Institute, Inc. and Indiana University School of Medicine, Indianapolis, IN

## Abstract

*Millions of people are either living with or dying from HIV/AIDS; most of this living and dying is taking place in developing countries. There is an immediate need for electronic medical record systems to help scale up HIV/AIDS prevention and treatment programs, reduce critical human errors, and support the research necessary to guide future efforts. Several groups are working on this problem, but most of this work is occurring within silos. To be more effective, we must find ways to collaborate. We describe a system built on the 30+ years of experience at Regenstrief Institute to serve as the seed for building toward a common infrastructure. We discuss the design goals, data model, and implementation of a data entry component. Further details are available online at [amrs.iukenya.org](http://amrs.iukenya.org).*

## Introduction

*"In the End, we will remember not the words of our enemies, but the silence of our friends."*

*-Martin Luther King, Jr.*

Our world is being ravaged by a pandemic of epic proportions. Millions of people are either living with or dying from HIV/AIDS. The vast majority (up to 95%) of this living and dying is taking place in developing countries. In sub-Saharan Africa, the hardest hit region, over three million people died from AIDS in 2004 alone. Most statistics suggest continued and unrelenting human decimation unless provocative, large-scale interventions are put into place. Silence is not an option.

Many have not been silent. Multiple missionary and humanitarian projects have taken root throughout Africa and international aid organizations have stepped in to provide resources and coordinate efforts.<sup>1</sup> However, these embryonic efforts face poorly developed health care infrastructures and a continually growing patient population. Health care providers are overworked and overwhelmed, often seeing upwards of 80 to 90 patients per day. Given man's inherent "non-perfectability", these environments are especially conducive to errors – errors with life and death consequences.<sup>2</sup> Anecdotal experience from physicians on the front line teaches us these mistakes are not just probable, they are

already occurring. As these care systems must inevitably be scaled up, more and more patient contact is being tasked to providers with less training, further increasing the likelihood of errors.

In the meantime, the treatment of HIV continues to evolve. As sites struggle to establish basic systems of care, opportunities to collect data and evaluate optimal treatment methods have too often gone unclaimed. Provocative research innovations are crucial to our long term success.

Designing strategies to address these immediate needs must focus on the efficient and accurate capture and transmission of clinical information. Incorporating well designed clinical data repositories into these sites would allow information to be leveraged across clinical, public health, and research arenas simultaneously. Within clinical care settings, just-in-time patient data facilitates the care process, and alerts/reminders enhance care quality.<sup>3,4</sup> These same data can facilitate public health epidemiological efforts and fuel both prospective and retrospective research projects.

Many sites devoted to improving health in developing countries have arrived at these same conclusions independently and are developing computerized systems of their own.<sup>5,6</sup> Consequently, the urgent need for handling information has led, in many cases, to disparate rapid developmental efforts. While the requirements from one context to another may vary, much of the foundational work is redundant. The results of these efforts often lack the ability to scale and are not standards-based, as resource and time constraints dictate designs which meet specific, immediate needs.

In hopes to reduce this redundancy and promote a scalable, flexible system built on standards, we describe and freely share an electronic medical record (EMR) infrastructure, built upon the experience at Regenstrief Institute. We have seeded collaboration with colleagues at Partners in Health (Boston, MA) and Columbia University. It is our hope that by contributing our vision, concept dictionary, tools, source code, and data model openly to the community, we will stimulate collaboration and encourage a sorely needed division of labor.

## Background

Indiana University School and Moi University School of Medicine (Eldoret, Kenya) have enjoyed a long and fruitful partnership over the past fifteen years. In February of 2001, this partnership led the first published electronic medical record system in sub-Saharan Africa, the Mosoriot Medical Record System (MMRS).<sup>7</sup> It was developed in Microsoft® Access and installed in a rural primary care health center in Western Kenya.

At the same time, in response to the growing number of patients suffering and dying from HIV/AIDS, the AMPATH (Academic Model for the Prevention and Treatment of HIV/AIDS) project was started and the MMRS was quickly adapted to serve as a database for the HIV clinics.<sup>8</sup> Over the next few years, thousands of patients and tens of thousand of visits were recorded into this database. By 2003, the productivity and vision of AMPATH had drawn the attention of the WHO, USAID, the Gates Foundation, and other international agencies. The initial two clinics had grown to eight different sites and plans were in place to grow further; the patient load increased and so did the demands on the database. The authors were asked to consult on how to scale the database to meet the growing demands of AMPATH. Our answer was AMRS.

Why not purchase a system or adopt an existing open-source system? While several open-source EMRs were available (e.g., [OpenEMR](#), [CARE2X](#), [VISTA](#), etc.), these systems either fell short of our requirement to serve both clinical *and* research needs – i.e., to collect non-ambiguous, coded data – or were not easily adaptable to our setting.

## Setting

The AMPATH project currently supports eleven clinic sites in Western Kenya. We have internet access via satellite at a central site and localized networks at several other sites. We are working toward linking all of the sites. Data capture is performed in the clinics on paper encounter forms. The encounter forms are brought to a central location for data entry, transcribed into the data repository, and returned to the clinic to be placed in the patient's paper chart. As sites are networked, we anticipate data entry will migrate to the remote clinic sites.

## Design Goals

Our design goals are outlined in Figure 1. We desire not only to serve the needs of the AMPATH project, but also create a foundation from which collaborative efforts can grow. At the foundation of this effort is the data model. Upon the model, we built the first

**Figure 1. Design Goals used for the AMRS**

<p><b>Collaboration</b> – systems need to be developed openly and built upon a common infrastructure, the sharing of “best of breed” modules can best occur within a shared infrastructure</p> <p><b>Scalability</b> – the infrastructure must not only handle thousands of patients and hundreds of thousands of observations, but also be scalable to tens of thousands of patients and millions of observations</p> <p><b>Flexibility</b> – systems must support not only HIV-centered care, but also general medical care, since clinical care is not limited to HIV care</p> <p><b>Rapid form design</b> – data collection needs are a moving target; therefore, form design and deployment must allow for continual change</p> <p><b>Clinically useful</b> – feedback to providers and caregivers is critical – if the system is not <i>clinically</i> useful, it will not be used</p> <p><b>Use of standards</b> – to maximize the flexibility and extensibility of the system</p> <p><b>Support high-quality research</b> – via non-ambiguous, coded data</p> <p><b>Web-based with support for intermittent connections</b> – <i>developing countries do not always have reliable power or internet connections, but when available, internet-based technologies offer increased scalability</i></p> <p><b>Low cost</b> – if the system is to be widely available and adaptable in developing countries, cost must not prohibit adoption. Ideally, the nuts and bolts of the system should be downloadable for free.</p>
--

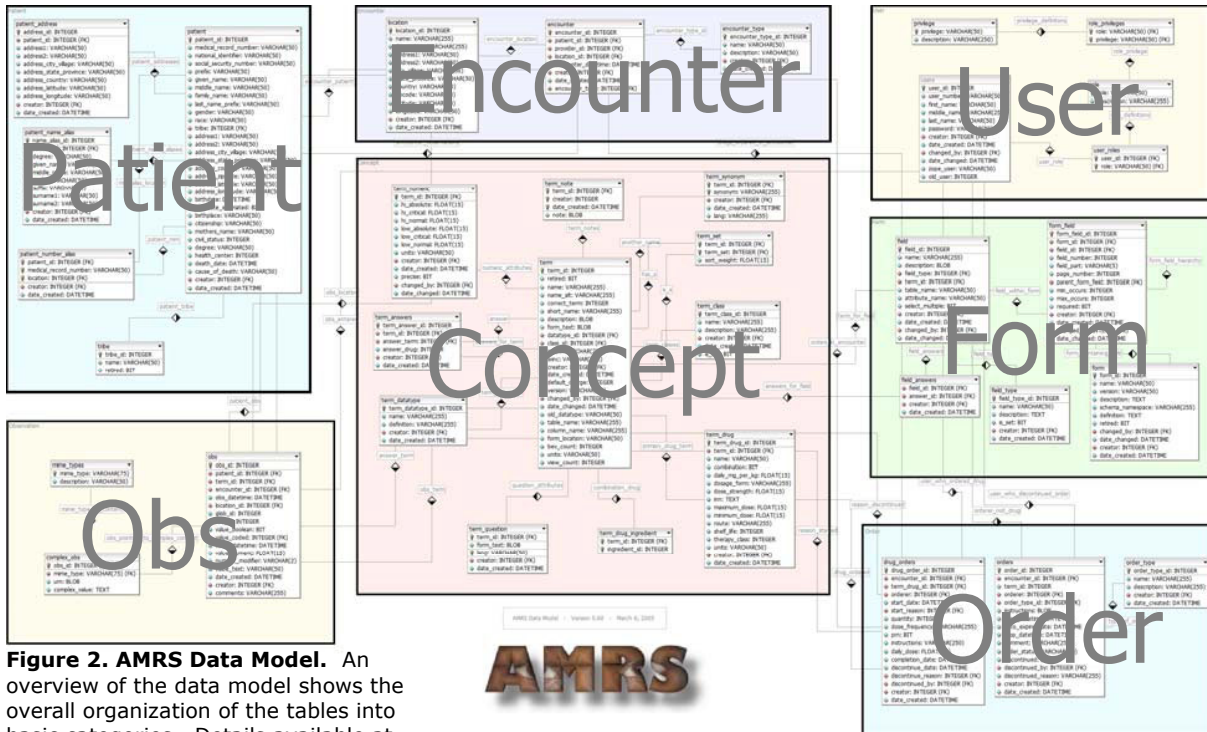
clinical application, a data entry module. First we introduce the data model, and then we describe the overall system design.

## Data Model

At the heart of any EMR lies the data model. The structure of the data model dictates the scalability and flexibility of a system. Knowing this, we designed ours using knowledge and experience gleaned from work with the Regenstrief Medical Records System. We found a useful, open-source tool, DBDesigner ([www.fabforce.net](http://www.fabforce.net)), which we used to build and manipulate the data model in a rich graphical interface (see Figure 2).

We enhanced the data model by collaborating with colleagues from Partners in Health, who have been doing similar work with TB and HIV in Haiti and Peru.<sup>9</sup> After connecting at MedInfo 2004, we discovered considerable overlap between our data models and began working toward a common design.

The core of this data model addresses the who, what, when, where, and how of medical encounters. For each of these basic categories, several tables are used to model the data.



**Figure 2. AMRS Data Model.** An overview of the data model shows the overall organization of the tables into basic categories. Details available at [amrs.iukenya.org/datamodel](http://amrs.iukenya.org/datamodel).

<b>Who?</b>	Patient, Provider (User)
<b>What?</b>	Concept, Observation
<b>When?</b>	Encounter
<b>Where?</b>	Location
<b>How?</b>	Order, Form

Two key design decisions greatly enhance the scalability of our system: a concept dictionary and a "stacked" method of storing clinical observations.

The concept dictionary contains all concept definitions as coded entities, including both questions and answers. For example, both the question CIVIL STATUS and the possible answers, SINGLE, MARRIED, DIVORCED, SEPARATED, etc., exist as unique concepts in the concept dictionary. Concepts are classified (test, drug, diagnosis, etc.) and assigned a data type (numeric, coded, boolean, etc.). Whenever possible, concepts are mapped to standardized vocabularies such as LOINC or ICD-10.

In many systems, including the MMRS, encounter data are stored within a "flat" table – i.e., one encounter per row. Therefore, if fifty observations are made during an encounter, the table must have at least fifty columns in which to store those observations. This "flat" model is very similar to a spreadsheet; it is easy to comprehend and can be

developed rapidly, but it lacks scalability. If fifty new observations are needed, the table structure must be altered to accommodate these new findings. To avoid adding columns, columns end up re-used over time and deciphering the data becomes a nightmare. Scaling to thousands of different observations becomes impossible.

For better scalability, we adopted the model used by larger systems; namely, an entity-attribute-value approach. In this model, a table of observations is used to store one observation per row. The "flat" row of observations for a particular encounter is literally turned on its side so the observations become stacked. Each individual observation becomes a row in the "obs" table, including the patient, encounter, concept being measured, reported value, and relevant timestamps. Using this model, extending the system with new observations no longer requires alteration of the model. Instead, a concept for the new observation is created in the concept dictionary along with, if appropriate, any coded answers that are new concepts to the system. Accommodating new or changed forms becomes relatively trivial.

For the latest version of the data model in use, please refer to our website at [amrs.iukenya.org/datamodel](http://amrs.iukenya.org/datamodel).

### System Design

We implemented our data model in MySQL® ([www.mysql.com](http://www.mysql.com)) and used Plone ([www.plone.org](http://www.plone.org)) to quickly create a web interface from which to edit

our concept dictionary. Plone is a web portal built on Zope ([www.zope.org](http://www.zope.org)) and primarily written in Python ([www.python.org](http://www.python.org)). For increased security, all Plone content is served through Apache's web server ([www.apache.org](http://www.apache.org)). The system can run on either Windows or Linux platforms. The Plone portal provides out-of-the-box user authentication, session-management, centralized documentation, file-sharing, group announcements, and a convenient template-based site design. Plone communicates to the database using SQL queries (ZSQL methods) through a Python-based ODBC bridge (mxODBC Zope Database Adapter, [egenix.com](http://egenix.com)).

We created concepts for the concept dictionary by modeling concepts directly off the paper forms used by the AMPATH project – we did this to ensure that the system would easily accommodate AMPATH form data. Given the concepts on a particular visit form, we group them in a form definition and use Python scripting on the website to dynamically generate an XML Schema representation of the form. This schema is loaded into a Web Forms tool. In our case, we selected Microsoft® Office InfoPath™ ([office.microsoft.com](http://office.microsoft.com)). While not open-source InfoPath™ provides very simple drag-and-drop WYSIWYG (what you see is what you get) interface for form design and doubles as a data entry tool capable of validating form data against an XML Schema before submitting it to our server. Other Web Forms technologies exist (XForms, Web Forms 2.0), but lacked the functionality we needed at the time. While InfoPath™ provides critical functionality for data entry, we purposefully minimize InfoPath™-specific coding in our system to allow switching to an equivalent open-source solution if one becomes available.

Data entry personnel log into the website, identify a patient, and select the appropriate encounter form. An XML template is dynamically generated which triggers InfoPath™ to open and load the appropriate

form definition off the server. As the data entry person progresses through the form, a task pane is used to provide field-sensitive choice lists, assistance, and notifications (see Figure 3).

By using web forms technology, we are able to provide data-driven forms that can be rapidly designed and easily adapted to changing needs. This flexibility comes from a clean separation between the form data and its appearance. Supporting multiple views of the same form data or altering the visual appearance of a form becomes trivial.

Once the encounter form is completed, the data enterer clicks on a submit button which send the form data in XML format to the server. The server uses an XSLT translation to convert the form data into an HL7 message, which is then sent to an HL7 listener (currently written in perl). The HL7 listener parses the HL7 message and populates the database. If any errors occur along the way, they are presented to the data enterer, who corrects the problem, and re-submits the form. Using the HL7 standard to accept data allows the system to accept input from other data sources without constant re-coding.

### Lessons Learned Thus Far

*Importance of a Concept Dictionary.* An essential core of an EMR is its concept dictionary. Those systems that lack clear definitions or a coded system for handling medical concepts will punish those using, maintaining, and/or trying to cull data from the system.

*Always return to the "practical".* When developing a system, it is important to stay close to what matters – i.e., the clinical care. Allowing your development to stray too far from immediate needs will lead to decreased productivity and/or functionality that nobody can use. For example, an elaborate method for performing patient searches based on demographic data may seem useful, but if those data are not routinely collected, the functionality serves

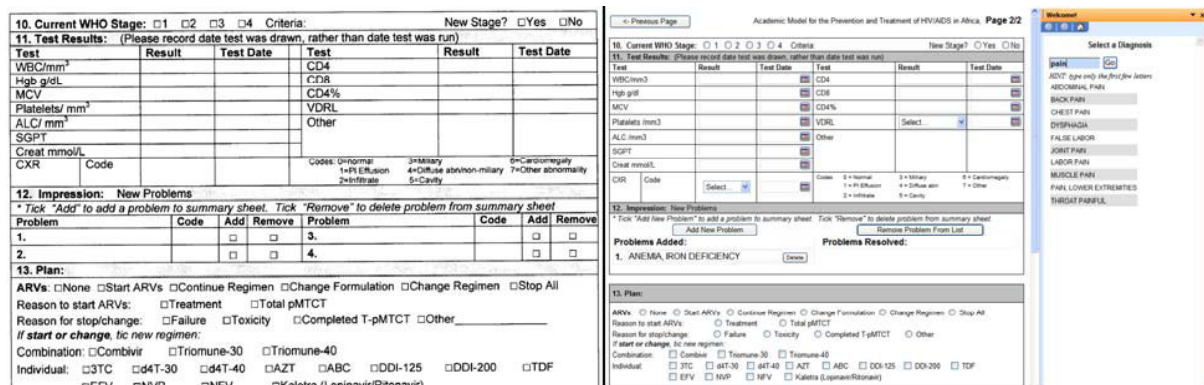


Figure 3. Data entry form. On the left, the original paper form; on the right, the electronic equivalent able to deliver data directly to the central repository.

little purpose.

*Build the dictionary based on need.* When creating a concept dictionary, we found it much more productive to work from existing forms, rather than trying to empirically add concepts that might be needed.

*Fight ambiguity.* Data integrity is quickly compromised when questions or responses to questions can be interpreted in more than one way. To maximize usefulness of coded data, concepts must be well-defined and unique within the dictionary. Working from the data entry forms, we uncovered ambiguous questions and were able to improve both the forms and the dictionary as we progressed.

*Easy & shareable tools.* Working from a website allows for easier collaboration. We initially developed the website for our own use, but we quickly found ourselves referring to the website when discussing concepts and design issues with the team.

*The fractal of data modeling.* Both for the data model itself and the modeling of concepts, we struggled with the line between under- and over-modeling. A useful rule of thumb that we routinely used was asking the question: “what is clinically useful?” For example, do you make a single concept for “right upper lobe,” two concepts “right” and “upper lobe,” or the three concepts “right,” “upper,” and “lobe of lung.” In our case, we found that these debates were best solved by making concepts only as granular as were clinically relevant (either for patient care or subsequent research).

### **Discussion**

We used primarily open-source and freely available tools to build our database and the data entry tool. The two exceptions are the ODBC adapter (modestly priced at \$95 USD) and Microsoft® Office InfoPath™, which is included as part of Microsoft® Office 2003.

Admittedly, there are strengths and weaknesses to our approach. Particular strengths of AMRS include the focus on coded data, its flexibility and scalability, and the use of standards. Limitations include the lack of a full suite of tools (i.e., appointment scheduling and billing), the learning curve needed to understand the data model, and the relative lack of Python expertise.

### **Conclusion**

While the AMRS data model and the data entry tool represent yet another EMR for public consumption,

we hope that the efforts we have invested in modeling and the use of standards along with our willingness to share and collaborate can raise the bar for EMR design in developing countries. When responding to the pandemic of HIV/AIDS, there is no time for anything less. Our millions of HIV-infected friends need to know that we will not be silent.

*“It seems essential, in relationships and all tasks, that we concentrate only on what is most significant and important.”*

*-Søren Kierkegaard*

Questions/Comments? Please visit our website at [amrs.iukenya.org](http://amrs.iukenya.org).

### **Acknowledgements**

*This project was supported by a grant from the Rockefeller Foundation and grant number 1-D43-TW01082 from the Fogarty International Center, National Institutes of Health. We are indebted to Dr. William Tierney for his leadership and to Dr. Joseph Mamlin, who has pioneered the AMPATH project.*

### **References**

1. Fleck F. WHO insists it can meet its target for antiretrovirals by 2005. *BMJ* 2004;329(7458):129.
2. McDonald CJ. Protocol-based computer reminders, the quality of care and the non-perfectability of man. *New England Journal of Medicine* 1976;295(24):1351-5.
3. Biondich PG, Anand V, Downs SM, McDonald CJ. Using adaptive turnaround documents to electronically acquire structured data in clinical settings. *AMIA 2003;Annual Symposium Proceedings/AMIA Symposium*.:86-90.
4. Dexter PR, Perkins S, Overhage JM, Maharry K, Kohler RB, McDonald CJ. A computerized reminder system to increase the use of preventive care for hospitalized patients. *N Engl J Med* 2001;345(13):965-70.
5. Tomasi E, Facchini LA, Maia Mde F. Health information technology in primary health care in developing countries: a literature review. *Bulletin of the World Health Organization* 2004;82(11):867-74.
6. Fraser HS, Jazayeri D, Mitnick CD, Mukherjee JS, Bayona J. Informatics tools to monitor progress and outcomes of patients with drug resistant tuberculosis in Peru. *Proceedings / AMIA 2002;Annual Symposium*.:270-4.
7. Rotich JK, Hannan TJ, Smith FE, Bii J, Odero WW, Vu N, et al. 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* 2003;10(4):295-303.
8. Voelker R. Conquering HIV and stigma in Kenya. *JAMA* 2004;292(2):157-9.
9. Fraser HS, Jazayeri D, Nevil P, Karacaoglu Y, Farmer PE, Lyon E, et al. An information system and medical record to support HIV treatment in rural Haiti. *BMJ* 2004;329(7475):1142-6.