HST 184: Health Information Systems to Improve Quality of Care in Resource-Poor Settings

Designing a Public Health Software Framework: Porting OpenMRS data to i2b2

Final Paper Submitted by:

Tewuh Fomunyam (credit)
Jamie Symonds (audit)
Stephen Lorenz (audit)

May 13, 2011

Advisers: Hamish Fraser and Evan Pankey

Introduction

Over the past several years the role of technology in resource poor settings has seen a rapid increase in innovation and development. These innovations have been spawned in part by the availability of open source software, cheaper hardware including laptop "servers" and cell phone technology, and a renewed interest in alleviating the suffering of those living in poverty. Three of the eight Millennium Development Goals (World Health Organization, 2010) directly address the importance of health care in eliminating poverty. Computing technology can play an important role in increasing the availability and improving the quality of health care in resource poor settings. While the use of such technology holds real promise, it also exposes weaknesses and challenges. One of these weaknesses is the difficulty in disseminating medical data from disparate systems for use in discovery research and analysis.

Accurate information is critical in improving the quality and sustainability of any system, whether the system is manufacturing, retail, education, or health care. Without accurate data it is difficult to measure how an organization or system is doing and where to make improvements (Fraser, 2011). For this reason, it is vital that the best analytical discovery tools are available to those striving to understand the strengths and weaknesses of systems in the hope of making improvements.

For our class project we developed a framework to simplify the adoption of one such analytical tool, i2b2. Typically the implementation of i2b2 is a complex and time consuming process, requiring significant time and resource commitment and an intimate technical knowledge of the underlying structure of the system. This makes the costs of implementation of i2b2 too steep for many organizations. The goal of our framework is to simplify the implementation of i2b2 to allow this tool to be more accessible to those working in resource poor settings for the improvement of health care. We used OpenMRS, an open source electronic medical record (EMR), as a reference system to develop our framework.

Background

Of course, in order to take advantage of any electronic data analysis tool, the data must be entered accurately into an electronic database. There are unique challenges to supporting electronic medical data entry in resource poor settings, such as the lack of a stable infrastructure to support computing equipment and technical staff (Blaya, 2010), resistance to change, and political forces. Analysis of these barriers to the adoption of electronic medical records, while critical, are out of the scope of this paper.

Electronic medical record systems are used primarily to maintain patient information as related to medical encounters, observations, and diagnoses. EMRs excel at supporting the day-to-day operations required to run a clinic, hospital, or health care system. EMRs are designed to be patient centric systems and are not optimized to support ad-hoc data discovery of potentially multi-source data from across an enterprise. A data discovery tool, such as i2b2, can enhance the reporting capabilities of an organization by providing optimized, de-identified, aggregate analysis of a population.

i2b2 is used heavily in research for quickly identifying cohorts of patients. While the reporting functionality of an EMR might be capable of identifying a given population, system administrators could be concerned and reluctant to grant researchers access to confidential patient data stored in an EMR. This kind of analysis can be off-loaded to an i2b2 instance without needing to grant researchers direct access to an EMR. The confidentiality of patient data is accomplished by “de-identifying” the patient medical information as it is imported into i2b2. Once in i2b2 researchers from the local clinic, to the ministry of health, to researchers in an academic institution on the other side of the globe can query the i2b2 database without compromising patient confidentiality.

OpenMRS

OpenMRS is a popular open source electronic medical record used in over 50 countries around the world (OpenMRS Atlas, 2011). OpenMRS has been an attractive choice for clinics and research institutions due, in part, to its open nature, low deployment costs, and ease of extending the system through modules.   In 2004 the Regenstrief Institute and Partners In Health established OpenMRS with the goal of creating an open source, community supported electronic medical record system.

OpenMRS consists of a Java web application, an API, and a database. The Java web application draws heavily on the third party libraries Spring and Hibernate. The Spring Framework was designed to simplify the development of Java programming by addressing the complexities introduced by the J2EE architecture.   Spring relies on dependency injection and annotations to greatly simplify and speed up the development of Java applications. Spring web applications follow the Model-View-Controller (MVC) design pattern for web development. Hibernate is an object-relational mapping tool used to map database records to Java objects ("Object-relational mapping," 2011). The OpenMRS API is designed to sit between the web application and the database and simplify access to the data model.   OpenMRS can run on any platform that supports Java technology, such as Linux, Windows, and Mac OS X. While OpenMRS is typically installed on top of a MySQL database, the application is considered database agnostic and can run with any standard relational database management system.

OpenMRS is built around a robust concept dictionary (Ball, 2011). The concept dictionary maintains all the possible questions and answers, or facts, that can be assigned to a patient. Concepts are divided into different classes, such as symptoms, diagnosis, lab tests and results, drugs, and procedures. OpenMRS concepts are also broken down by data type. Some data types include numeric, coded, text, and Boolean. The question/answer combination is considered an observation in OpenMRS and would be stored in the obs table. The obs table maintains the crosswalk between the questions and the answers. For any given patient visit, or "encounter," recorded in OpenMRS, many observations will be collected about the patient.

Observations can be simple questions and answers represented by concepts. Consider the question: "Are you experiencing night sweats?," and the answer: "No". The question would be stored in OpenMRS as a concept. The answer, "No," is also a concept and would be considered a "coded" data type. Not all answers are concepts. For example, with the question "CD4 Count" there are hundreds of possible answers. For this kind of question a numeric data type, such as "350," would be appropriate.

Because all facts about a patient, excluding basic demographic information, are built upon concepts and are mapped to patients through observations allows OpenMRS to be flexible and support internationalization. Issues such as the localization of the application is possible because the concept dictionary abstracts the text of questions and coded answers. For example, the concept representing "cough" might have the concept id of 452 in OpenMRS. The French word for cough, "toux," or the Cameroon Metta dialect word for cough, "kwe," would all share the same concept id, 452. (Ball, 2011)

While the concept dictionary in each OpenMRS instance is autonomous, it is possible to map local concepts to standard medical vocabularies, such as SNOMED-CT, LOINC, ICD-9, and RxNORM. These vocabularies would be considered “concept sources” in OpenMRS. The mapping of concepts to external vocabularies improves the compatibility of OpenMRS data with other medical systems and makes it easier to generate an ontology, or hierarchical representation, of the data. This is valuable in creating a fully functional i2b2 instance.

i2b2

i2b2 (Informatics for Integrating Biology and the Bedside) is a scalable informatics framework that enables usage of existing clinical data for discovery research. It is an open source Java-based project with an i2b2 web client and a Rich Client Platform (RCP) workbench application. i2b2 is built as a modular system or "hive", with many “cells,” some of which are core cells. There are optional cells, clients, and plug-ins that are used to extend the core functionality.

i2b2 allows you to build patient cohorts through a graphical user interface; the client is tree-based, referencing an hierarchical ontology, which enables the user to build a dynamic query, drilling down through layers and applying filters. The ontology is mapped to a vocabulary in the underlying dataset from the source electronic medical records.

Typical use cases for this type of analytical software include "driving biology projects" such as identifying asthma, rheumatoid arthritis, IBD (Inflammatory Bowel Disease), MS (Multiple Sclerosis), and diabetes, as exemplified by our interview with two i2b2 team members from Partners HealthCare (Gainer, 2011). This team uses an in-house tool known as the "data mart builder" that creates the data mart in the i2b2 star schema, which is a series of pre-built custom SQL statements used to retrieve, arrange and upload the source system data to an i2b2 data mart.

An interview with Recombinant, a company that assists in implementing i2b2 solutions, makes the distinction between data warehouse and data mart, with a data mart being a smaller database with one specific purpose. Their implemented solutions typically consist of a data warehouse, with clinical systems (EMRs), departmental systems, billing systems, pharmacy systems, and scheduling systems feeding the warehouse, and building several purpose-built data marts from the warehouse data. Tools such as Kettle are used to load data. (Palchuk, 2011)

Recombinant reiterates the former use case already identified, that i2b2 has a narrow use case which is to find out if you have a population or a cohort that you can include in a study. i2b2 excels at answering the question, "How many patients are there?" Being able to answer such a question on-the-fly can lead to new research opportunities and funding sources. This kind of question, when drawing from multiple data sources, is often too time consuming to tabulate manually.

Another use case noted is that of hypotheses testing, most notably being able to compare a site's patient population with a suspected hypothesis, where decisions can be made concerning populations on the basis of sample information. i2b2 has also lent itself to essentially being in EMR in some implementations, used as a registry, and used for quality improvement purposes.

An interview with Cincinnati Children's Hospital (Marsolo, 2011) reiterated the typical i2b2 use cases; using i2b2 to identify cohorts of patients where users query a de-identified data mart to produce a matching list of records. In addition, Cincinnati Children's Hospital uses i2b2 to analyze outcome measures with their clinical effectiveness group and medical records operations. Cincinnati Children's Hospital also references a patient registry use case which is useful in consolidating patient data from multiple systems, including legacy systems (Marsolo, 2011).

Interoperability between i2b2 instances is possible, however the i2b2 instances must be based on the same ontology. SHRINE is an i2b2 plug-in, or “cell,” that is used to run “federated queries.” Federated queries can run across multiple instances of i2b2 hosted at different locations across the Internet. SHRINE is useful only when aggregate data is needed across the instances.

i2b2 officially supports two database types, Oracle and Microsoft SQL Server. However it might be possible to run on an open-source database, like MySQL. There may be one instance of i2b2 running on an open-source database, and some have endeavored to achieve this, however in most cases here in the United States, the institutions implementing i2b2 already have available Oracle and/or Microsoft SQL Server licenses. More research needs to be done to determine if i2b2 can be deployed on an open-source database, such as MySQL. This would be useful for organizations working in resource poor settings where licensing costs could be prohibitive to adoption. i2b2 has been customized to run on the JBoss application server on either Windows or Linux platforms.

As demonstrated from the use cases presented, the core strength of i2b2 is in the ad-hoc identification of patient cohorts. i2b2 forces the organizations to start thinking of the secondary uses of clinical data and that data from point-of-entry clinical software can serve multiple purposes. (Palchuk, 2011)

Research Questions

Our research questions for this project are:

- Can we create a framework to make the adoption of i2b2 more accessible for organizations operating in resource poor settings?

- What best practices, if any, exist for importing data from source systems, such as OpenMRS, into i2b2?
- What standards should we follow to protect patient health information (PHI)?
- How can we create an meaningful ontology from the “flat” OpenMRS concept dictionary?
- What standards and data formats should we consider for transporting data between source systems and i2b2?

Methods

To answer the above questions, we explored the current literature on OpenMRS and i2b2 to better understand both systems. Next we conducted personal interviews with developers of OpenMRS and i2b2 to learn more about the structure of each system and the challenges we should anticipate as we set out. We also interviewed users of i2b2 to get a better understanding of the typical use cases for i2b2 and how that might benefit organizations who are actively using OpenMRS.

We had a chance explore both OpenMRS and i2b2 to familiarize ourselves with the user interfaces and the back-end databases. To explore OpenMRS we used de-identified data from the Partners In Health Malawi OpenMRS implementation. To better understand the i2b2 data model we used the default demo data from the i2b2 database. We set a timeline and had regular weekly meetings with our advisers to brainstorm ideas and design the structure of our framework.

Analysis

As we considered potential designs for our framework, one of our main objectives was to develop a process that moved i2b2 within reach for organizations operating in resource poor settings. We aimed for as close to a turn-key solution as possible. Whatever framework we designed needed to adhere to the following stipulations: it must 1) be open source, 2) where possible utilize open standards, 3) be data source agnostic, and 4) be simple to implement without compromising core functionality.

Developing an open source system allows future developers to build on what we have developed, as well as, debug our code. Where possible we followed open standards, to reduce the learning curve for other developers and to allow our programs to be as interoperable as possible. Our solution will be data source agnostic, as well. While we are using OpenMRS as our reference EMR for this project, we want our framework to be able to support data from any electronic medical system, whether it be an EMR, laboratory, inventory, or pharmacy system. Finally, our framework needs to strike a balance between simplicity, where we potentially compromise on functionality, and complexity, where we provide complete control at the expense of loosing potential adopters due to a steep learning curve.

Ultimately, we discovered that the best way to develop our framework was to try to implement the framework in experimental, proof-of-concept stages, using an agile development model ("Agile software development," 2011). This process revealed some weaknesses in our framework. For example, early on in the project we considered leveraging the Continuity of Care Record (CCR) ("Continuity of Care Record," 2011) standard to move data between source systems and i2b2. As we started to look deeper and attempt to develop the code to generate CCR, we discovered that this standard was not appropriate for representing many of the elements required by i2b2. Working within an agile development model we were able to adapt accordingly and investigate other solutions.

Our reference source system was an OpenMRS instance loaded with de-indentified data from the Partners In Health Malawi system. We started by developing a standalone Java application which migrated OpenMRS data to i2b2 using a basic SQL-to-SQL process. This step was useful in helping us more deeply understand the table structures of OpenMRS and i2b2. See Appendix A for a diagram which visually maps OpenMRS and i2b2 data structures.

The i2b2 database is comprised of several schemas, several of which can be considered system schemas. The system schemas support the i2b2 application cells but do not contain medical data. The remaining schemas are populated with data from the source systems. The metadata schema is roughly analogous to the OpenMRS concept dictionary and contains the ontologies required to support i2b2. i2b2 can support an arbitrary number of ontologies within one i2b2 instance. i2b2 also supports a schema which stores the patient, concept, encounter, provider, and observation data within a special database design called a star schema ("Star schema," 2011). The i2b2 star schema contains a fact table, called observation_fact, and four dimension tables: concept_dimension, patient_dimension, provider_dimension, and visit_dimension. Data integrity is maintained in the star schema through database constraints. The star schema is optimized for the efficient querying of vast amounts of data. See Appendix B for an illustration of the i2b2 star schema.

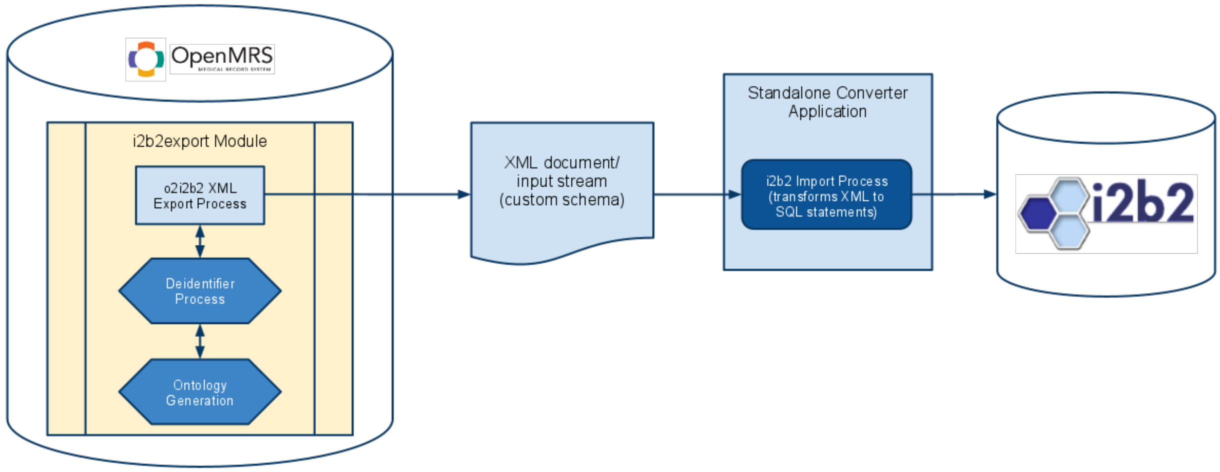
Despite the fact that both OpenMRS and i2b2 share many of the same elements, such as concepts, patients, encounters, providers, and observations, the migration is not as straightforward as it might seem. It is necessary to prepare the OpenMRS data prior to export so that it can be imported into the i2b2 star schema. For example, two critical transformations need to be applied to the OpenMRS data on export. These are to de-identify the data to protect patient health information (PHI) and to map OpenMRS concepts to an i2b2 ontology. We use the "safe-harbor" method (National Institutes of Health, 2004) to de-identify personal health information, by replacing patient and encounter identifiers with new incremental identifiers. For identifying dates, such as birthdate, death date, and encounter dates, we randomly generate the month and day while preserving the year. For patients over 89 years of age we chose a random birthdate within 2 years of the actual age. We map the generated patient and encounter identifiers with the actual identifiers within the OpenMRS database. This identifier mapping, which remains in OpenMRS and is not exported, will allow researchers to track down individual patients, if necessary, provided they receive IRB approval.

The other important transformation that must be made to OpenMRS data on export is the mapping of concepts to an ontology. An ontology, in i2b2 terms, is a hierarchical representation of concepts. The idea of an ontology is one of the core strengths of i2b2. Without an ontology all concepts would exist at the same level and it would be difficult to query for classes of concepts. For example, consider the two diagnoses of "Abdominal aortic aneurysm" and "Atherosclerosis of aorta." These two diagnoses both fall under the following ontology: Diagnoses -> Circulatory System -> Arterial vascular disease. Within this ontology, the "Abdominal aortic aneurysm" concept falls under the sub-level "Aortic aneurysm and dissection" and the "Atherosclerosis of aorta" concept falls under "Atherosclerosis". This ontology allows users to easily search either for patients with a specific diagnosis, such as "Abdominal aortic aneurysm," or for patients with any malady of "Arterial vascular disease." Users can also step up through the ontology to query any patients with a disease of the circulatory system.

In our proof-of-concept implementation for this class, we mapped OpenMRS concepts to a more basic ontology, organized by: OpenMRS implementation name -> concept classes -> concepts. We took this approach because developing an ontology is a time consuming process involving input from many areas of an organization. In the future, we hope to use the database behind the Maternal Concept Lab project (“Maternal Concept Lab,” 2011) to help us map local OpenMRS concepts to standard vocabularies such as SNOMED-CT and RxNorm. From there we should be able to generate a meaningful and robust ontology from the OpenMRS data.

We explored several design scenarios in the process of developing our framework which are listed along with companion diagrams in Appendix C. The design we decided on uses a custom XML schema as the medium through which source system data is loaded into i2b2. We have developed an OpenMRS module which simplifies the generation of the XML for organizations using OpenMRS. The module includes several options including the ability to limit data export by patient cohort, only export new patients, encounters, and observations, and to specify whether or not to use an internal ontology or to map concepts to a standard ontology, where possible.

Framework Design Diagram


OpenMRS
i2b2export Module
o2i2b2 XML Export Process
Deidentifier Process
Ontology Generation
XML document/ input stream (custom schema)
Standalone Converter Application
i2b2 Import Process (transforms XML to SQL statements)
i2b2

We also created a standalone Java application which transforms the XML document into SQL statements and imports the data into the i2b2 database. This i2b2Import application can optionally clear an entire i2b2 instance of stale data. This is useful if you want to periodically refresh your i2b2 instance with new data.

This framework meets all of our objectives. It strikes a balance between ease of use and unlimited functionality. It is data source agnostic so it can work with any source system so long as the XML schema is followed. All code is open source and we adhered to open standards, where possible.

Conclusion

This project has helped us better appreciate the complexities of creating an open source health care framework for use in resource poor settings. We deepened our understanding of OpenMRS, i2b2, and the different standards and tools involved in porting medical data between systems. In addition, issues such as the confidentiality of patient information, understanding data warehousing technology, and how i2b2 can augment the reporting and discovery capabilities of an organization, have been central to our project.

While we have made considerable progress in the development of this framework, we have more work to do and questions to answer. For example, it would be helpful to solicit more feedback from OpenMRS implementers to gauge their interest and requirements for an i2b2 implementation. In addition, more research into alternative, open source databases, such as MySQL, for i2b2 would be helpful for organizations working in resource poor settings.

As of this writing, we have a robust design for our framework and we have started to develop the tools required to support its implementation. These tools include a new OpenMRS i2b2 Export Module and an i2b2Import application. We also created an XML schema definition (XSD) to support the migration of medical data from any source system into i2b2. We plan to continue to develop and promote these tools after the conclusion of the course. Once the OpenMRS module is complete, we will host the module at http://modules.openmrs.org and create an OpenMRS wiki page to document the module and the project generally. On June 16, 2011 we will present on our project on the weekly OpenMRS developers conference call.

The framework and tools we have developed hold promise for making i2b2 more accessible for organizations with limited resources. In particular, organizations using OpenMRS might find these new tools especially helpful and could use i2b2 to augment their ability to perform data discovery and clinical research. We hope our project will have an impact on the quality and delivery of health care for those living in resource poor settings.

Acknowledgements

Our interviewees and advisers were particularly generous with their time and expertise and deserve honorable mention in our paper. We would like to thank several members of Partners In Health, including Rita Cuckovich, Ellen Ball, Evan Waters, and Hamish Fraser. Our first project interview was with Rita, and she adeptly introduced us to the OpenMRS Reporting Module and the idea of patient cohorts. Ellen Ball shared her extensive knowledge of the OpenMRS concept dictionary and also helped extract and de-identify the Malawi dataset used in several group projects in the class. Evan Waters also contributed his time and expertise of the PIH Malawi dataset answering several rounds of questions from project teams. Our project, let alone the course, would not have been the same without Hamish's expertise, insight, and guidance.

We would also like to thank Evan Pankey, one of our team advisers, for his diligence, experience, and encouragement throughout the project. His deep knowledge of development projects, project management, and his lecture on conducting "goal directed research" were helpful in thinking of strategies to conduct interviews and develop our framework.

Two i2b2 team members, Vivian Gainer and Nich Wattanasin from Partners Healthcare gave us our first real understanding of i2b2 and best practices for implementation. Their expertise and patience were helpful as we just started to grasp the scope of our project more deeply. Matvey Palchuk of Recombinant Data Corp shared his extensive understanding of i2b2 and medical informatics with our team. His articulate explanations of the strengths and weakness of i2b2 and of possible use cases were helpful as we came to better understand the utility of i2b2. Matvey's wide experience and perspective in the field of medical informatics were refreshing and

enlightening. Finally, we would like to thank Keith Marsolo of Cincinnati Children's Hospital who gave us a most comprehensive overview of i2b2 use cases. Keith's name and Cincinnati Children's Hospital kept coming up in our earlier interviews as a hands-on i2b2 expert; our expectations were not disappointed.

The contributions of time and expertise from these individuals were invaluable to our project conception and design. Thank you.

References

World Health Organization (2010, May). “Fact sheet N°290: WHO | Millennium Development Goals: progress towards the health-related Millennium Development Goals, http://www.who.int/mediacentre/factsheets/fs290/en/index.html

Fraser, H. S. F. (2011, Feb.). “Design and impact of health information systems in developing countries.,” HST.184: Health Information Systems to Improve Quality of Care in Resource-Poor Settings [Lecture], MIT, Cambridge, Massachusetts.

Blaya, J. A., Fraser, H. S.F., & Holt, B (2010). Implementing Medical Information Systems in Developing Countries, what Works and what Doesn’t. AMIA 2010 Symposium Proceedings.

Star schema. (n.d.). In Wikipedia. Retrieved March 16, 2011, from http://en.wikipedia.org/wiki/Star_schema

Object-relational mapping. (n.d.). In Wikipedia. Retrieved May 8, 2011, from http://en.wikipedia.org/wiki/Object-relational_mapping

National Institutes of Health, (2004, Jan. 12). “HIPAA Privacy Rule and Its Impacts on Research,” http://privacyruleandresearch.nih.gov/research_repositories.asp

Cuckovich, R. (2011, March 18) of Partners In Health. Personal interview regarding OpenMRS Reporting Module with Tewuh Fomunyam, Jamie Symonds, and Stephen Lorenz.

Gainer, V. S. & Wattanasin, N. (2011, March 23) of Partners Health Care. Personal interview regarding i2b2 with Tewuh Fomunyam, Jamie Symonds, and Stephen Lorenz.

Ball, E. (2011, March 23). Personal interview regarding OpenMRS with Tewuh Fomunyam, Jamie Symonds, and Stephen Lorenz.

Palchuk, M. (2011, March 29) of Recombinant Data Corp. Personal interview regarding i2b2 with Tewuh Fomunyam and Stephen Lorenz.

Marsolo, K. (2011, April 11) of Cincinnati Children's Hospital interview regarding i2b2 with Tewuh Fomunyam and Stephen Lorenz.

OpenMRS Altas, (n.d.). In OpenMRS. Retrieved February 5, 2011, from http://openmrs.org/about/locations.

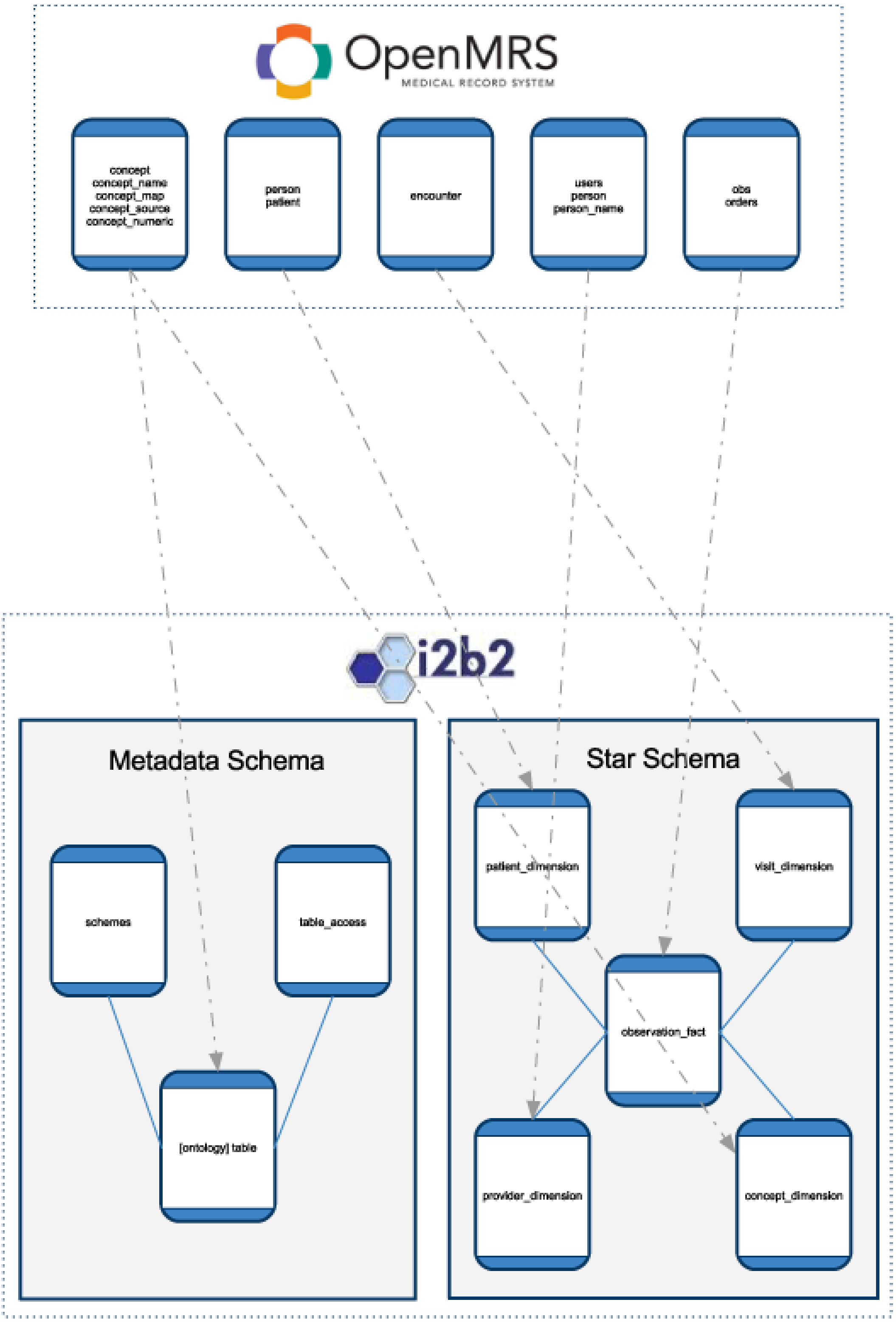
Agile software development, (n.d.). In Wikipedia. Retrieved May 8, 2011, from http://

en.wikipedia.org/wiki/Agile_software_development

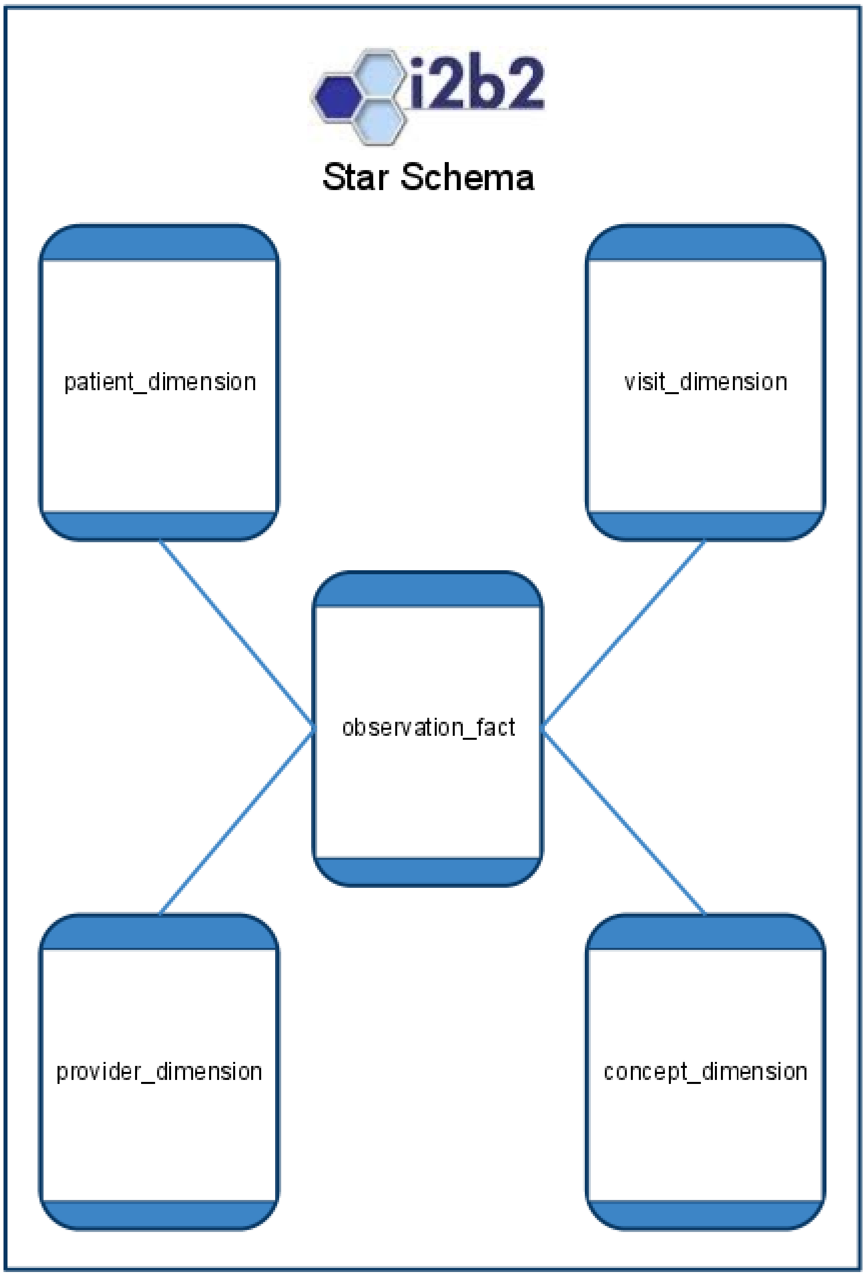
Continuity of Care Record (n.d.). In Wikipedia. Retrieved March 10, 2011 from http://en.wikipedia.org/wiki/Continuity_of_Care_Record

Maternal Concept Lab, (n.d.). In Maternal Concept Lab Wiki. Retrieved May 8, 2011, from http://www.maternalconceptlab.com/wiki/Main_Page

Appendix A - OpenMRS to i2b2 Data Mapping:

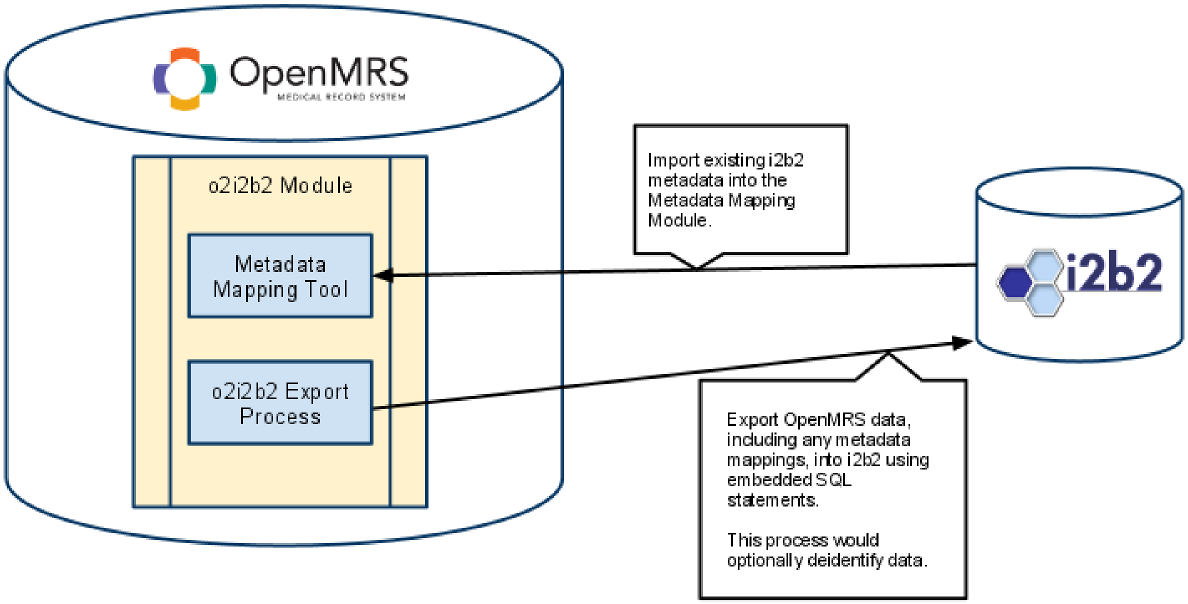


Appendix B - i2b2 Star Schema:



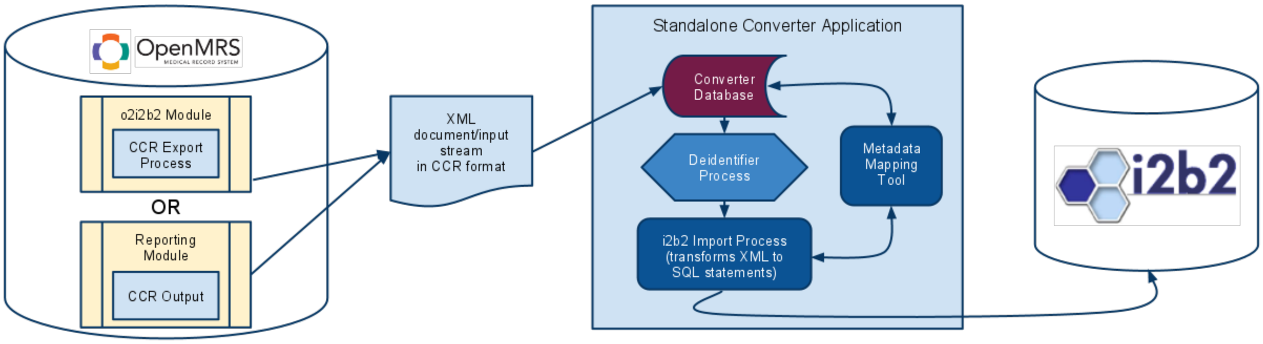
Appendix C - Design Scenarios:

Scenario 1: OpenMRS Module


OpenMRS
MEDICAL RECORD SYSTEM
o2i2b2 Module
Metadata Mapping Tool
o2i2b2 Export Process
Import existing i2b2 metadata into the Metadata Mapping Module.
i2b2
Export OpenMRS data, including any metadata mappings, into i2b2 using embedded SQL statements.
This process would optionally deidentify data.

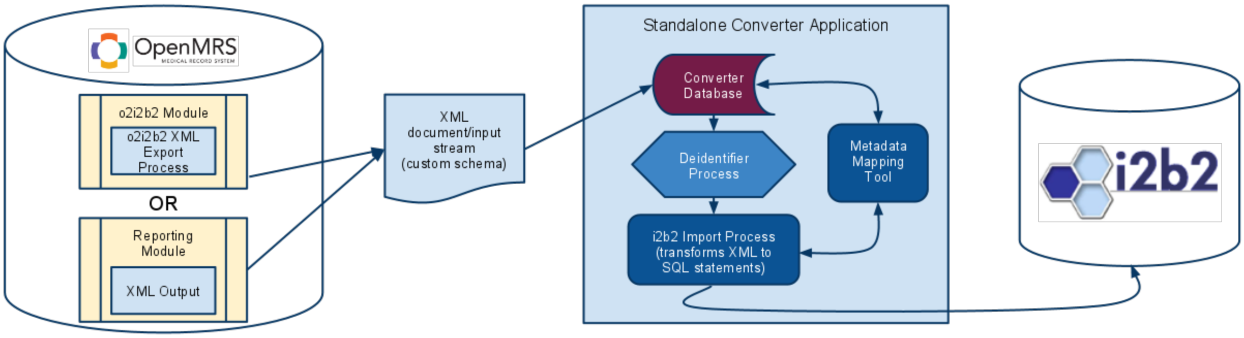
Pros:	Cons:
Easy for OpenMRS users to implement	Not really a "framework"
Transfer data to i2b2 in one step	Specific to OpenMRS

Scenario 2: Continuity of Care Record (CCR) with Standalone Application



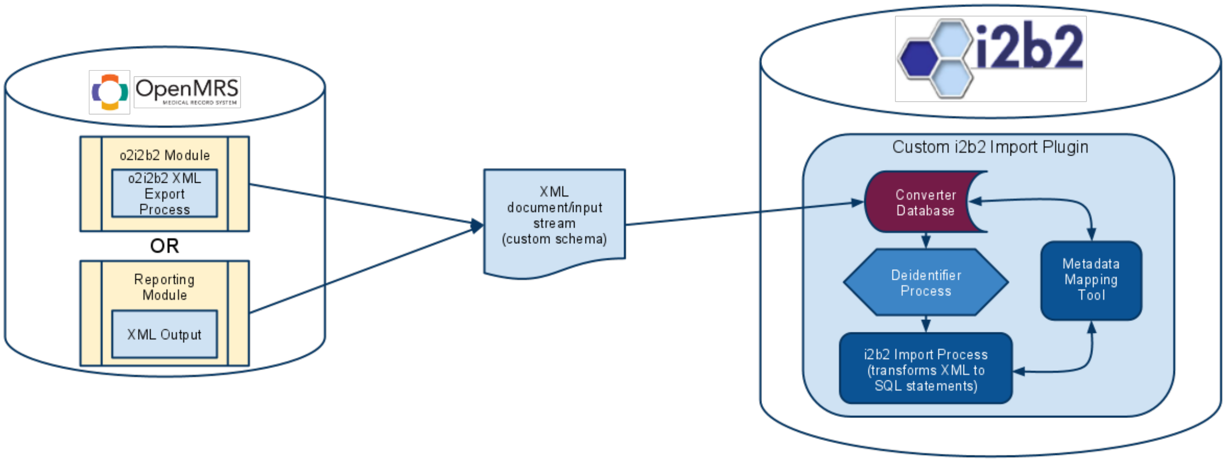
Pros:	Cons:
Built upon a current standard	CCR does not support the detail required by i2b2
Systems might already export CCR formatted data	Verbose

Scenario 3: Custom XML with Standalone Application


OpenMRS
MEDICAL RECORD SYSTEM
o2i2b2 Module
o2i2b2 XML Export Process
OR
Reporting Module
XML Output
XML document/input stream (custom schema)
Standalone Converter Application
Converter Database
Deidentifier Process
Metadata Mapping Tool
i2b2 Import Process (transforms XML to SQL statements)
i2b2

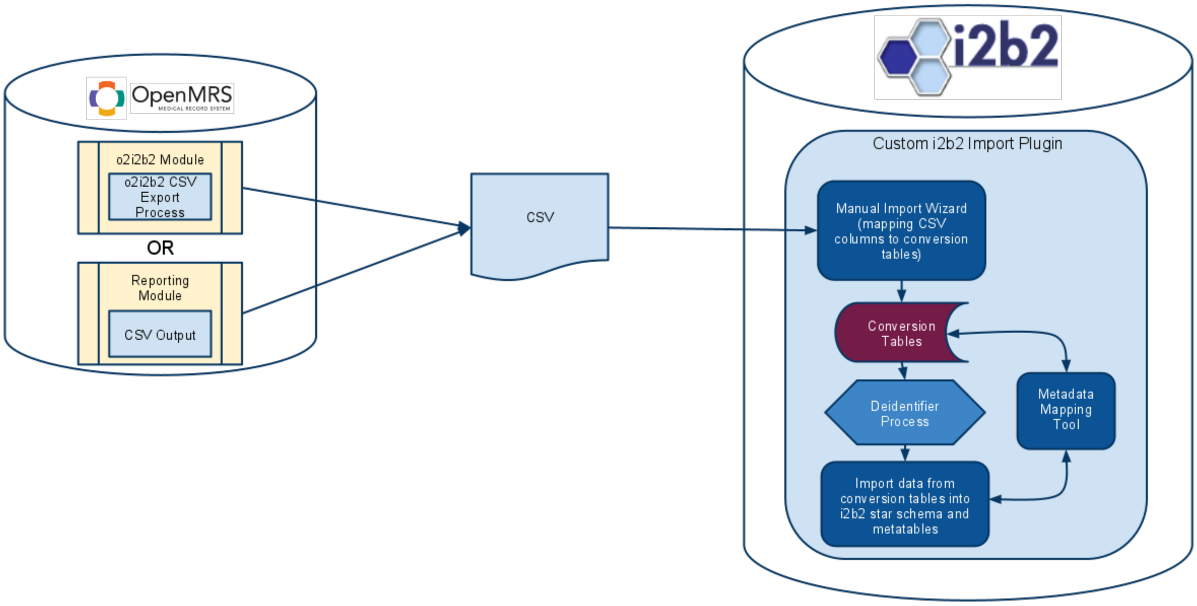
Pros:	Cons:
An extensible framework	Lots of “moving parts”
Source agnostic: should work with any EMR, Lab, Pharamcy system	Yet another XML schema to learn

Scenario 4: Custom XML with Custom i2b2 Import Plug-in


OpenMRS
MEDICAL RECORD SYSTEM
o2i2b2 Module
o2i2b2 XML Export Process
OR
Reporting Module
XML Output
XML document/input stream (custom schema)
i2b2
Custom i2b2 Import Plugin
Converter Database
Deidentifier Process
Metadata Mapping Tool
i2b2 Import Process (transforms XML to SQL statements)

Pros:	Cons:
An extensible framework	Lots of “moving parts”
Source agnostic: should work with any EMR, Lab, Pharamcy system	Yet another XML schema to learn
	Significant developer learning curve to develop

Scenario 5: CSV with Custom i2b2 Import Plug-in


OpenMRS
o2i2b2 Module
o2i2b2 CSV Export Process
OR
Reporting Module
CSV Output
CSV
i2b2
Custom i2b2 Import Plugin
Manual Import Wizard (mapping CSV columns to conversion tables)
Conversion Tables
Deidentifier Process
Metadata Mapping Tool
Import data from conversion tables into i2b2 star schema and metatables

Pros:	Cons:
Easy to export CSV even from legacy systems	Lots of “moving parts”
Source agnostic: should work with any EMR, Lab, Pharamcy system	No enforced schema
Faily compact	Requires manually mapping CSV columns to conversion columns
	Prone to human error

Appendix D - Useful Tools:

Oxygen XML (http://www.oxygenxml.com): Useful for manipulating and viewing XML and XSD files.

Inqscribe (http://www.inqscribe.com): Interview transcription software with the ability to easily pause, slow down, or step back through an audio file while transcribing. We used the 30 day free trial.

Free Conferencing (http://www.freeconferencing.com): A free online phone conferencing tool which supports dial-in, Skype, and conference call recording. Useful for our weekly meetings and phone interviews.

Dropbox (http://www.dropbox.com): Cloud based file sharing tool (think FTP for the new millennium) which integrates with your desktop environment and makes file sharing seemless.

Google Docs (http://docs.google.com): Cloud based word processing tool used to collaboratively write class papers.

VMware Player (http://www.vmware.com/products/player): Free utility which allows you to run VMware virtual machines on your computer. Useful for testing i2b2 demo virtual machine.

MySQL (http://www.mysql.com): Free, open-source database used to hold sample OpenMRS database.

Eclipse (http://www.eclipse.org): Open source, community based development platform used in our class for Java programming.