Interoperability Solutions through AI Managed Generic Data

James MacDougall

2550 S York Street, Apt#301

Denver, CO 80210

James.macdougall7@gmail.com

An article entitled “America’s Shocking Maternal Deaths”, appeared in the New York Times on September 4th. It brought to the public eye an alarming issue that maternal death rate in the USA is far higher than most first world countries. This is an issue I was introduced to by Doctor Steven Hasley of Magee-Womans Hospital of UPMC. He like many in his field has been aware of the problem but lacked the information needed to discover the underlying causes. The fundamental problem with gathering the feedback data is the variety of data standards and health care systems which are currently storing the needed information. Fortunately there is now incentive from the government to allow open API’s to access the data, also the Office of Population Affairs has asked for a set of 34 data points to be returned for every patient who is issued title ten funding. To take advantage of this new opportunity, Dr. Hasley envisioned an open API which used FHIR standards and OAuth 2.0 security to query health records to build the OPA’s desired data, then later a more comprehensive maternal data set. I volunteered as a computer scientist with a background in AI and data mining to write the program to gather this data. This project is ongoing, the team expanded to include Viet Nguyen and Susan Matney who specialize in FHIR and CIMI modeling respectively. However, as this project has been impeded by the lack of FHIR compliant health care systems it has familiarized me with the problems surrounding the lack of interoperability between health care systems. As a data miner it’s become clear that a purely FHIR based approach won’t fix the interoperability problem but my specialization with AI has provided the tools I believe can dramatically improve the situation.

Currently there are a number of factors impeding widespread interoperability of health care systems. The most fundamental, is that the systems are designed with different purposes in mind, from accounting to medical modeling, to disease tracking, etc. The different goals help feed the next biggest issue, the wide variety of standards for representing and sending data. While many will use standards like ICD-10, LOINC, SNOWMED there are roughly 4000 different systems with a profuse amount of human entered data. In addition the data standards are in a constant state of evolution, even working with just FHIR there are three distributions to manage. These issues dramatically impair the ability of consumers to share their data with different providers often necessitating faxing of files or retesting. It also impairs providers and researchers who would like to access the data for analysis. The project I am

proposing is a way to bridge such gaps in the information flow. The target is to aid consumers and providers with slightly different implementations of the same core programming.

This program will use Java programming because it can run on most platforms including mobile phones and because the implementation needs the powerful programming tools native to Java. This also allows it to be deployed online as an Open API which UPMC is planning to host for their data gathering. The basic concept is the program will take a data input given to it by a server or other source (FHIR, Excel, raw). Then it will run an analysis on the raw data to recognize the format, pulling the data from the input source into its own internal data structure. From there it can then be viewed, lasting changes made to the mapping between data structure and ultimately sent out in another format.

For a consumer a version of this can be released on app stores and online which only holds a single persons data. They could then populate the program with their own data either through manual entry or sent by a provider in an unknown format. The program data will be kept encrypted for security reasons. UMA 1.0 Standards will be implemented to give user the ability to securely send complete or de-identified data to their provider or researcher and the program will handle mapping the data into the format required by the next user.

For a health provider the same program could be used to help enter data in their system but for research purposes a more advanced system will be needed. In particular it will need to be compliant with OAuth 2.0 the users getting the correct permissions to gather data. There will need to be a data structure to hold numerous patients data which comes in from different sources. Finally the export feature will be limited to exporting an encrypted excel file with all the patient data for analysis. No analysis is meant to occur in the program but their needs to be more organization options for the creation of the excel file.

The most important part is obviously how to translate data from one standard to another, it's also the most complex so please consult the slides for a visual aid. Data translation in handled by the interaction of several complex data structures. Every basic piece of data from a person is held by the program in a dataPoint object which is a sophisticated polymorphic object. At minimum it contains the actual data as well as list of possible titles (SNOWMED, LOINC, etc.) that the data falls under. It will also always contain a list of the other dataPoints that are part of the same patient resource allowing an individual patient's data to be tracked, appended or have identifying data suppressed without compromising the data structure. The dataPoints are organized by templates which also serve as a registry for factory object for the dataPoints and a means to save mappings. A template when saved is a list of instructions for factories of certain types of dataPoints, plus rules to be applied on how to recognize or transmit a specific data format. Fuzzy logic, an algorithm for determining the odds of a similar data being matching, is built into the dataPoint isEqual functions. This helps increase flexibility while mapping between templates but all matches are approved by the user who can input specific rules into a template to override dataPoint actions. The setup is very generic allowing any form of data to exist in a dataPoint so long as it has knowledge about the category of object to which it belongs. Category objects are also initialized by the templates and help dataPoints get organized by the information they hold. The templates also allow the user to create and save custom mappings for

categories complete with added syntax to interact with other data standards. Ideally most of the mapping work is handled by AI. I've implemented my own fuzzy logic algorithm in the current version but in future I'm considering the potential of SyntaxNet a google natural language neural network to improve the handling of human inputs. Using these techniques the program, when it cannot recognize the format of a new data source, will build its own template with a suggested mapping for the user. The feedback setup is somewhat similar to neural networks though reliant on humans for error catching. As the templates get more use they will gain more information about the standards they are translating and need less human interaction. This is not a complete summary of the necessary data structures but a description meant for non-programmers.

This application's success will be measured by its ability to accurately pull in data from a variety of sources. The existing testing resources include several examples of FHIR servers as well as some raw and excel sources. Using these sources the program's success will rely on its ability to recognize the given source automatically and with minimal human intervention accurately assign the data to the correct categories. Templates to recognize FHIR data, raw and excel data are currently in progress and each template will be treated as a modular expansion to the program. The number of functional modules will determine the flexibility of the program and allow different versions to be released to interact with different data standards. This also allows smaller implementations of the program to be considered it some data standards prove resistant to mapping. The ability of researchers to analyze the excel document it outputs will be an important measure of success for the provider. In all cases the ability to correctly output an internal data source into the desired FHIR or custom format will be assessed.

Once complete the provider version is already wanted by Magee's Woman's hospital which will assess whether they can use it to gather maternity data from a variety of providers. I know that basic versions of this program are possible as I've spent the last several months creating one for FHIR standards. As the only programmer currently familiar with this project I will be handling all the coding. However, frequent usability tests and conferences to work on this project are scheduled between me and Dr. Hasley plus associates. I expect a basic version to take 3 months to create between my other obligations. Most progress after that will be focused on training the program and expanding the standards that it recognizes. This done primarily for my own edification while I pursue mastery in AI, as such the funds will go to freeing the time it deserves in my schedule from other jobs. If the provider version passes testing I will be giving it to UPMC and using their connections to expose it to other providers. It will also be marketed online through stores such as the upcoming HSPC app store.

A consumer version will be released for a variety of mobile platforms through the existing app stores for android, iPhone etc. Titanium Appcelerator is a proven program for distributing a program across multiple mobile platforms and doing the underlying programming in java allows for a great deal of flexibility. I'm also hoping that as it becomes useful for providers more consumers will be informed by them of the potential benefits to being in control of their own data.

Making health care systems communicate to each other requires a level of interoperability. Numerous mistakes and delays are caused by the inability of health care systems to send each other

data. Individually mapping all the data from one standard to another is a time consuming process that could be easily outdated by changing standards. Programming in AI functions at a fundamental level will give the program an advantage in keeping up with the changing standards with less human effort. Such applications are rare partially because of the complexity/rarity of AI programming and the fact the only recently did common computers get powerful enough to handle such algorithms. This is meant to be a very limited program but very good at what it does, allowing users to handle any data standard with the same ease.