

Creation of a Technical Tool to Improve Syndromic Surveillance Onboarding in Tennessee

Jeffrey Leegon*, Caleb Wiedeman, Vamshi Krishna Nukala and Paul Petersen

Surveillance, Tennessee Department of Health, Nashville, TN, USA

Objective

To show how the creation of a software tool and implementation of new processes improved the efficiency of syndromic surveillance onboarding at the Tennessee Department of Health.

Introduction

Syndromic surveillance is commonly supported by information generated from electronic health record (EHR) systems and sent to public health via standardized messaging. Before public health can receive syndromic surveillance information from an EHR, a healthcare provider must demonstrate reliable and timely generation of messages according to national standards. This process is known as onboarding. Onboarding at the Tennessee Department of Health (TDH) focused heavily on human review of HL7 messages. However, the visual inspection of messages was time-intensive and delayed efforts to provide constructive feedback to participating healthcare providers. To ease the quantity of manual review done during the onboarding process, TDH created an application to assist in the process of reviewing syndromic surveillance messages.

Methods

The application for reviewing syndromic surveillance messages was developed in Python 3.6, a general purpose programming language. Python was selected because of the strong libraries already developed in the language for data analysis, database interaction, and interacting with healthcare related data. To support TDH onboarding efforts, the application performed three tasks: file handling, HL7 processing, and database loading.

File handling was completed using Python core libraries. Healthcare facilities participating in onboarding regularly uploaded test HL7 batch files containing all emergency department (ED) visits to a secure File Transfer Protocol (sFTP) server owned by the State of Tennessee. Files are then retrieved from the sFTP server and delivered to the TDH integration engine, Rhapsody. Rhapsody processes the incoming files and makes a copy available to the Python application. The copies are then loaded by the application into a database and backed up in an archive.

After the application has finished handling the received files, the raw HL7 messages within the files are processed to extract relevant information needed to validate the message. The extraction was supported by the “python-hl7” library. The application referenced a CSV file with the names and locations of all of the PHIN 2.0 guide HL7 data elements to guide data processing. Processed HL7 fields and file metadata were then extracted into a relational database using the Python library SQL Alchemy.

Results

The resulting relational database contained two tables. The first table captured the file metadata: filename, date and time when the file is loaded, message count, age of the file, and if the data have been deleted. The second data table contained selected data elements from the raw HL7 messages. These data were kept in a singular table and database view for each healthcare provider participating in onboarding was created. Maintaining a separate view for each healthcare provider allowed for specific time periods to be monitored

and analyzed based on when the healthcare provider has implemented message changes. Additionally, arrival times for message updates to data elements such as discharge diagnosis, race, and ethnicity were logged in the database tables.

Conclusions

The creation of this application has greatly assisted in the process of reviewing syndromic surveillance messages at TDH by automating data extraction and organization for all received HL7 messages. The database tables of extracted HL7 data elements allowed for easy analysis in any tool that can connect to a database (e.g., SAS, R, Python, Excel, Tableau) and faster, more manageable message validation. TDH has been able to communicate better feedback to more healthcare providers because of the efficiencies gained after deploying the application.

Keywords

syndromic; surveillance; programming; python; Automation

Acknowledgments

The authors would like to acknowledge John Paulett, the creator of the easy-HL7 library in Python, for use of the library and support during the development of this program.

*Jeffrey Leegon

E-mail: jeffrey.leegon@tn.gov

