INTERFACE ENGINE FOR HL7 MESSAGING

MINOR PROJECT REPORT

SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR

THE AWARD OF THE DEGREE OF

BACHELOR OF TECHNOLOGY

Information Technology

Submitted By:	Submitted To:
Aditya Sharma(1805484)	Prof. Ranjodh Kaur
Gourav Khera(1805509)	Assistant Professor
Kulvir Singh(1805521)	Major Project Coordinator

Department of Information Technology

Guru Nanak Dev Engineering College,

Ludhiana-141006

Abstract

The Standard is written from the assumption that an event in the real world of healthcare creates the need for data to flow among systems. The real-world event is called the trigger event. For example, the trigger event a patient is admitted may cause the need for data about that patient to be sent to a number of other systems. The trigger event, an item is used from floor stock on behalf of a patient, may cause the need for information about the patient and the item used to be sent from the patient care system to the patient accounting system and materials management system. When the transfer of information is initiated by the application system that deals with the triggering event, the transaction is termed an unsolicited update. HL7 allows the use of trigger events at several different levels of data granularity and inter-relationship. For example, most ADT trigger events concern single objects (such as an admit event, which creates a message that contains data about a single person and/or account). Other ADT trigger events are concerned with relationships between more than one object (e.g., the merge events, which specify patient or account merges). Some ADT trigger events pertain to a collection of objects that may have no significant interrelationships (e.g., a record-oriented location-based query, whose response contains data about a collection of inpatients who are related only temporarily by local geography).

ACKNOWLEDGEMENT

We are highly grateful to Dr. Sehijpal singh, Principal, Guru Nanak Dev Engineering College(GNDEC), Ludhiana, for providing this opportunity to carry the minor project work at making 'Interface Engine for HL7 Messaging Using Python'. The constant guidance and encouragement received from our mentor Dr. kulvinder Singh Mann in carrying out the project is really greatful. Our mentor encouraged us to make this project not only for college assigned work but for our own knowledge also. We express gratitude to our other teachers Prof. Hanit Karwal and Prof. Mohanjit Kaur Kang who selected our project and gave us the permission to make. At last we are indebted to all others whosoever has helped us in this project.

Aditya Sharma

Gourav Khera

Kulvir Singh

Contents

1 Introduction

1.1 Introduction to Project Page

The Control/Query chapter of this Standard defines the generic rules that apply to all messages. Subsequent sections define functionally specific messages to be exchanged among certain applications. The specific aspects of message definition that are addressed herein are: a) the form to be used in functional chapters for describing messages. This includes their purpose, their contents, and the interrelationships among them. This form is called an abstract message definition because it is purely a level 7 (application) definition. b) the HL7 encoding rules for converting an abstract message into a string of characters that comprise an actual message. c) the programming procedures required to exchange messages using the HL7 specifications. d) the anticipated relationship with lower level protocols. e) certain message segments that are components of all messages. f) a single message, the Acknowledgement message, that may be used unchanged in multiple applications.

1.2 Objectives

HL7—Standard for exchanging healthcare data Health Level 7 is a standard development organization. It develops specifications to structure, encode and exchange patient healthcare information. The underlying objective is to facilitate exchange between healthcare systems.

1.3 Identification/Reorganization of Need

a) Minimising duplication of work and improving clinical efficiency:- Patients often have many touch points with healthcare providers in their lifetime. For instance, an encounter at a GP could result in a referral to a physio, medication from the pharmacy, lab tests and might result in hospitalisation.

In instances like this, patients cannot be expected to carry around their entire health record with them. Standards like HL7 make more data available to healthcare providers and enable them to make better clinical decisions more efficiently to provide the best care for their patients.

b) Maximise value from EHRs:- To maximise value from EHRs interoperability is crucial. Having visibility over data that has been generated elsewhere, can be hard to achieve, sometimes impossible if systems expose information in a proprietary way that makes sense only to them.

c) HL7 standards act as a bridge between modern healthcare services and advancing information technology.

d) Compliance to HL7 standards increases the number of contexts and services that can be included in aggregating the modern longitudinal patient record.

1.4 Proposed System

The aim of this project is to give user the ease of data handling in medical field. As there are billions of records or data of patients or of doctors. To save this data in a standard form, this is the interface engine made in Tkinter using Python in which the graphical interface is there, where one can input the data or fileds and gets the output in a HL7 standard.

1.5 Unique Features of the System

Patients often have many touchpoints with healthcare providers in their lifetime. For instance, an encounter at a GP could result in a referral to a physio, medication from the pharmacy, lab tests and might result in hospitalisation.

In instances like this, patients cannot be expected to carry around their entire health record with them. Standards like HL7 make more data available to healthcare providers and enable them to make better clinical decisions more efficiently to provide the best care for their patients.

HL7 standards act as a bridge between modern healthcare services and advancing information technology.

Compliance to HL7 standards increases the number of contexts and services that can be included in aggregating the modern longitudinal patient record.

As a reflection of the healthcare shift towards the adoption of emerging data types and devices, the standards have been evolving to maintain their relevance and defining the rules to support new use cases. From traditional messaging and document-based paradigms to the most current APIs.

2 Requirement Analysis and System Specification

2.1 Feasibility Study (Technical, Economical, Operational)

The project is executed within a one semester frame. The time frame allocated for this project is limited but adequate through proper time management and planning. The aim of this project is to design an open source, easy-to-use interface HL7. 1).ECONOMIC FEASIBILITY and TECHNICAL FEASIBILITY--This project is based on an OS machine with low-cost hardware such as intel i3 Processor and 3gb RAM.

2).OPERATIONAL FEASIBILITY-This programme will feature a simple, user-friendly interface that will appeal to anyone with a basic understanding of how to operate an any OS machine.

2.2 Software Requirement Specification

1).DATA REQUIREMENT- The data we used is randomly generated inside our code. the output we get after successful execution of the code is saved in localhost notepad. Hence, no worries of data loss. Overall, the data is managed.

2).FUNCTIONAL REQUIREMENT- The functionalities that a developer must incorporate into software to accomplish use cases are referred to as functional requirements. These functions will be built in such a way that vendor independent system-to-system communication is possible.

3).PERFORMANCE REQUIREMENT-If the system is not connected, it must not add more than two seconds to the time it takes to accomplish an action. There must be no more than a ten-second delay in the logging of researcher data to the research centre. The speed with which directives are provided to the system will be affected by the efficiency of the software code.

4).MAINTAINABILITY REQUIREMENT-The system is as simple to use as feasible, with all capabilities accessible.

5).SECURITY REQUIREMENT-As in this project there is no involvement of internet, so the security concerns related to data breach are less..

6).LOOK AND FEEL REQUIREMENT-Modularity is incorporated into the system's architecture.

2.3 Validation

To validate the performance of our system a hardware implementation is done.For the hardware implementation(manual scenario), we can simply enter the required fields in the interface engine to get desired results.

2.4 Expected hurdles

In general, most standards organizations are broadly open, as is the case with the Internet Engineering Task Force (IETF). The IETF is responsible for many of the core internet standards and has no formal paid membership model required to participate. The organization is run by volunteers and is funded by meeting fees, sponsor ships, and by its parent organization, the Internet Society. HL7, on the other hand, is a membership organization. The organization requires health systems, software vendors, and other stakeholders to pay significant membership fees to participate in the feedback process outlined in Step One and to vote on changes to the standard. Furthermore, outside of healthcare, most standards are implemented completely — that is to say, 100% — or not at all. That binary provides simplicity by reducing the number of decision points. With HL7, however, there is a large amount of variance between how vendors implement HL7 standards in Step Two above. HL7 standards themselves are often architected to have a strong amount of variance due to the evolution of the HL7 organization. That variance is shaped by HL7's initial pre-World Wide Web use as a standard for exchange between different vendors within the same data center administered by the same IT staff at a health system. HL7 wasn't originally designed for the Internet; it was made for data exchange within the four walls of the data center in a hospital's basement. Given its original intent, maintaining compatibility to exchange information between health systems wasn't a high priority. Step Three, deployment to healthcare organizations, requires a significant amount of work. Deployment often requires a specific IT project along with the requisite funding, IT staff time, and technology infrastructure to support a new standard. Each implementation is highly customized for the healthcare organization, and deployment requires technology teams to be on-premise. For example, the patient's race is typically represented as an abbreviation or just a number that is unique to that health system. At one health system, 3 might represent Caucasian while at another health system it represents Pacific Islander. While this lack of consistency is business as usual for the healthcare industry, it's not how the process works in many other industries. When the Bluetooth Special Interests Group authors a new specification, for instance, vendors such as Apple and Bose implement the standard in their devices and that's it — there's no additional configuration needed to move from vendor implementation stage to using the standards. (Headphones owners might need to occasionally update the software on their phones or other devices, but the update runs automatically and requires no implementation or configuration decisions.) Additionally, different organizations have different priorities for how up-to-date they keep their systems, and the overall landscape becomes fragmented in terms of both the version adopted and health system-specific configurations. In Stage Four, the frequency of health system-specific configuration of HL7 standards presents interesting challenges to application developers. Each health system will have implemented a customized set of specifications, and vendors seeking to work with those health systems traditionally need to adhere to each custom setup.

2.5 OSI Model to be used

The following figure shows the simplest graphical representation of the OSI model which we have followed:-

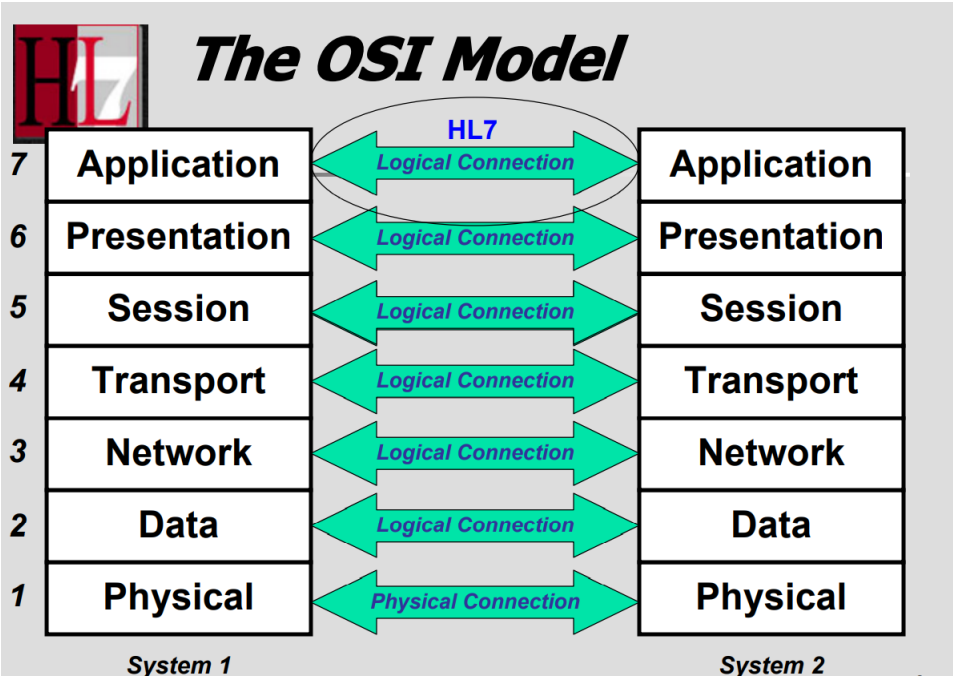

Figure 1: Graphical Representation of OSI Model

The application level addresses definition of the data to be exchanged, the timing of the interchange, and the communication of certain errors to the application.The seventh level supports such functions as security checks, participant identification,availability checks, exchange mechanism negotiations and, most importantly, data exchange structuring.

3 System Design

3.1 Design Approach (Object oriented)

The focus of the object-oriented approach is on encapsulating the structure and behaviour of information systems into tiny modules that mix data and process. The basic goal of Object Oriented Design (OOD) is to make system analysis and design more accessible in order to increase quality and productivity. In analysis phase, OO models are used to fill the gap between problem and solution. It performs well in situation where systems are undergoing continuous design, adaption, and maintenance. It identifies the objects in problem domain, classifying them in terms of data and behavior. The Object Oriented model is beneficial in the following ways: 1. It facilitates changes in the system at low cost. 2. It promotes the reuse of components. 3. It simplifies the problem of integrating components to configure large system. 4. It simplifies the design of distributed systems.

3.2 User Interface Design

The user interface is based on Tk GUI toolkit. A simple and interactive way of interaction between user and the design. A person must have a knowledge of how to run a program. It can be accessed anywhere anytime. The output or data it produces is saved on a local-host on which this design works. The interface we made is shown as below:-

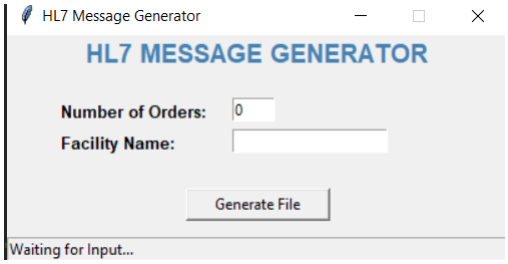

Figure 2: GUI

3.3 Methodology

1. First we need to download and import necessary packages like Tkinter, Random, Numpy packages.

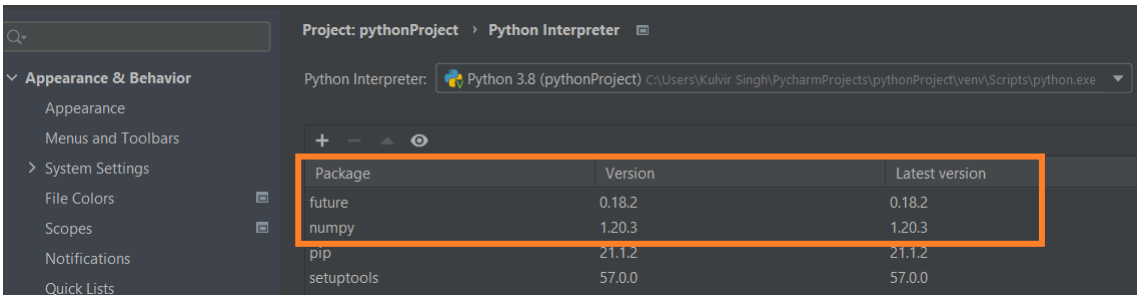

Figure 3: Installing Pacakges

2. Then define different functions which are to be used in the Tkinter part of the code like def setName, setDate, setMRN, setLableID, setDrug, most important the format of HL7 message i.e. HL7Message.

```
# Randomly pick and assign first and last name
def setName():
    listFirst = ['Bob', 'John', 'Adam', 'William', 'Jason', 'Ken', 'Kishan', 'Jessica', 'Olivia',
                 'Emma', 'Amber', 'Mia', 'Emily', 'Sarah', 'David', 'Isaac', 'Leah', 'Steve',
                 'Mohammad', 'Madison', 'Victoria', 'David', 'Henry', 'Ian', 'Nick', 'Jennifer',
                 'Hector', 'Aaron', 'Mike', 'Steve', 'Juan', 'Leeroy', 'Ashley', 'Heather']
    listLast = ['Smith', 'Johnson', 'Williams', 'Davis', 'Brown', 'Jones', 'Miller',
                'Wilson', 'Jackson', 'Thompson', 'Robinson', 'Lewis', 'Lopez', 'Carter',
                'Henderson', 'Peterson', 'Howard', 'Wood', 'Ramirez', 'Russell', 'Cox',
                'Diaz', 'Hill', 'Moore', 'Martinez', 'Green', 'Reed', 'Patel', 'Humphrey']
    firstName = random.choice(listFirst)
    lastName = random.choice(listLast)

    return firstName, lastName

# Set the start and end dates
# YYYYMMDDHHMMSS
def setDate():
    currentDate = datetime.datetime.now()
    ndate = str(currentDate)
    junk = ['-', ':', ' ']
    ndate = ndate.translate({ord(x): '' for x in junk})

    # Remove the seconds
    sec = '.'
    ndate = ndate.split(sec, 1)[0]

    # Add 5 mins to start date
    ndate = int(ndate) + 500
    edate = ndate
```

Figure 4: Defining Different Functions

3. By connecting the different functions to the MAIN one and to form a Tkinter window so that it will be easy interaction for the model.

```
#############################################################################
# TKINTER
#############################################################################
root = Tk()
root.title("HL7 Message Generator")
root.geometry("400x180+100+100")
root.resizable(0,0)
heading = Label(root, text="HL7 MESSAGE GENERATOR", font=("arial", 15, "bold"), fg="steelblue").pack()
subheading = Label(root, text="Internal Use Only | v1", font=("arial", 7), fg="steelblue").place(relx=0.99, rely=0.15, anchor=NE)

# Vars
global count_ent
global facility_ent
global status_ent
count_ent = IntVar()
facility_ent = StringVar()
status_ent = StringVar()

# Fields
Label(root, text="Number of Orders: ", font=("arial", 10, "bold")).place(x=40, y=50)
Entry(root, textvariable=count_ent, width=5).place(x=180, y=50)
Label(root, text="Facility Name: ", font=("arial", 10, "bold")).place(x=40, y=75)
Entry(root, textvariable=facility_ent).place(x=180, y=75)

# Buttons
but_generate = Button(root, width=15, text="Generate File", command=mainapp).place(relx=0.5, rely=0.75, anchor=CENTER)

# Note
# Label(root, text="", font=("arial", 7), fg="steelblue").place(relx=0.99, rely=0.91, anchor=SE)

# Status Bar
status = Label(root, textvariable=status_ent, bd=1, relief=SUNKEN, anchor=W)
```

Figure 5: Tkinter Code

4. This is the Output Window when we run the code, in it we fill the required fields and get the desired results.

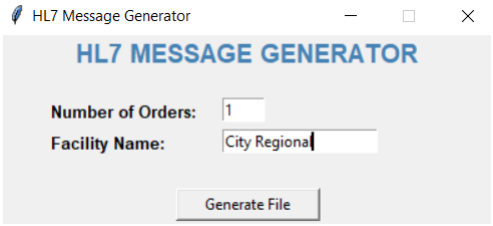

Figure 6: Inputting Values

5. The file is created at the given path. Generated .txt file containing HL7 Version2 message which looks like this:-

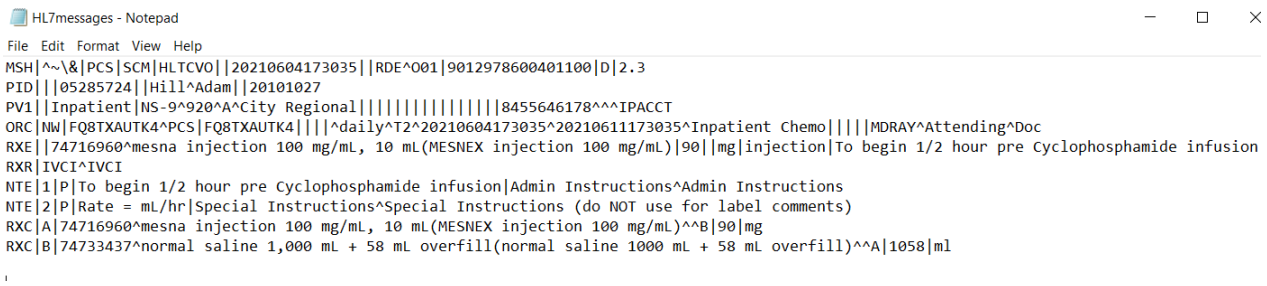

Figure 7: Output

4 Implementation

4.1 Introduction to Languages, IDE's, Tools and Technologies used for Implementation

PYTHON PROGRAMMING LANGUAGE :-

Python is an interpreted high-level general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant indentation. Its language constructs as well as its object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.Python is dynamically-typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library.

WHY LEARN PYTHON LANGUAGE :-

Python was built with the goal of getting rid of the complex and keeping only the necessary. Because of this, Python is easier to read, write, and learn than most other major programming languages.WP Engine surveyed 909 developers across the US on which programming language is the easiest to learn. Python came in second place after HTML, which some developers will argue is more of a scripting language than an actual programming language anyway.Python was noted for its "high readability and simple syntax that is easy to learn." The article also states that, "for new coders especially, the consistency and simplicity of Python both contribute to its ease of use and approachability."

IDE'S AND TOOL USED :-

PyCharm is an integrated development environment (IDE) used in computer programming, specifically for the Python language. It is developed by the Czech company JetBrains (formerly known as IntelliJ). It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems (VCSes), and supports web development with Django as well as data science with Anaconda. PyCharm is cross-platform, with Windows, macOS and Linux versions.

FEATURES OF PyCharm IDE :-

Coding assistance and analysis, with code completion, syntax and error highlighting, linter integration, and quick fixes Project and code navigation: specialized project views, file structure views and quick jumping between files, classes, methods and usages Python refactoring: includes rename, extract method, introduce variable, introduce constant, pull up, push down and others Support for web frameworks: Django, web2py and Flask [professional edition only] Integrated Python debugger Integrated unit testing, with line-by-line code coverage Google App Engine Python development [professional edition only] Version control integration: unified user interface for Mercurial, Git, Subversion, Perforce and CVS with change lists and merge Support

for scientific tools like matplotlib, numpy and scipy [professional edition only]

Tool used for making interface working is the Tk GUI tool, provided by Tkinter which is GUI library in Python. This library can be imported in PyCharm(IDE). It creates the GUI application main window. Add one or more of the above-mentioned widgets to the GUI application. Enter the main event loop to take action against each event triggered by the user.

4.2 Coding standards python language

Indeed coding and applying logic is the foundation of any programming language but there's also another factor that every coder must keep in mind while coding and that is the coding style. Keeping this in mind, Python maintains a strict way of order and format of scripting.Following this sometimes mandatory and is a great help on the user's end, to understand. Making it easy for others to read code is always a good idea, and adopting a nice coding style helps tremendously for that.

For Python, PEP 8 has emerged as the style guide that most projects adhere to; it promotes a very readable and eye-pleasing coding style. Every Python developer should read it at some point; here are the most important points extracted for you :-

1. Use 4-space indentation and no tabs.
2. Use docstrings.
3. Wrap lines so that they don't exceed 79 characters .
4. Use of regular and updated comments are valuable to both the coders and users.
5. Use of trailing commas.
6. Use Python's default UTF-8 or ASCII encodings and not any fancy encodings.
7. Use spaces around operators and after commas, but not directly inside bracketing constructs.
8. Characters that should not be used for identifiers.
9. While naming of function of methods always use self for the first argument .

5 Results

We set up and taught a graduate course to students with major in healthcare and in information technology. A total number of 76 students undertook the work, over several semesters. Students' background included several engineering branches in addition to non-engineering such as biochemistry, physiotherapy, kinesiology, and radiology technology. Students with background in civil engineering as well as non-engineering students had fewer skills in computers as compared to the others. However, they had other skills such as project management, experience with patients' interactions, or healthcare settings. One assistant provided supervision during nine hours distributed over few weeks, to help students in their work.

We asked the students to provide their HL7 messages for evaluation. We also asked them to analyze and comment on the communication received from the server, as responses to their messages.Students worked on problems from real clinical scenarios.

The first problem is inspired from a common real radiology process. By working on this problem, students not only dealt with almost all information and data flavors involved in radiology but also had the chance to touch the DICOM standard. Most importantly, they could grasp how the HL7 and DICOM standards fit one relative to the other, how they complement each other, and how they work together. Furthermore, the skills students acquired went beyond learning one or multiple standards: they learned how standards can cooperate to achieve an integrated workflow. Integration in medicine is still a big challenge [14, 15], at the conceptual level and at the implementation one, as well. Becoming familiar with these challenges and with existing integration solutions is one of the most important skills students must acquire.

The second problem is inspired from the EHR where querying for patient's identifier is a common need. EHR deployment has taken place in several countries. Its deployment is expected to expand as the quality of care is known to improve with EHR. Moreover, by reducing the duplication of tests and by improving the productivity, EHR is expected to increase healthcare efficacy and efficiency.

The interactions implemented as part of this problem are amongst the basic interactions needed to implement an EHR. In fact, a physician normally needs to access all the patient's diagnostic information. However, part of this information may have been generated in a different hospital using a different patient's identifier. Therefore, mapping several identifiers belonging to the same patient is a common need. On the other hand, the patient's identifier is not always known. Instead, using only a subset of the patient's demographic information for identification, such as using his/her name and/or birth date, is also a common need.

Therefore, skills learned from this work encompassed the HL7 standard and extended to reach the introduction to EHR sub-systems, their interactions, and the healthcare distributed process in general. Skills included several aspects related to HL7 such as constructing messages, understanding HL7 responses and acknowledgements, and getting familiar with typical HL7 interactions and workflows, as well as understanding

the HL7 documentation structure and its navigation. Skills also included several IHE profiles and actors such as understanding the roles of the IHE PDQ supplier and PDQ consumer, the IHE PIX manager, and PIX consumer, as well as being introduced to the IHE documentation. Students also acquired skills related to XML and its schemas, an important tool in the domain of information technology.

6 Conclusion and Future Scope

6.1 Conclusion

We have presented a new method to actively learn HL7 rapidly, by solving real clinical problems. We have shown that working with real information objects and constructing messages that could have been exchanged in real clinical contexts enabled effective learning of our complex subject. We have also shown how solving a real problem can accelerate the learning and motivate the browsing and the searching of information in the very large standard documentation. In fact, to solve the practical problems, students browsed the standard documentation: to find the answers they needed, to construct the message, to find out and understand the message content and structure, and to find out and understand specific data types along with their meaning.

Concentrating on a specific problem directed the students to focus on a subset of the huge data model, helping them grasp that subset in detail without getting lost or intimidated by the size of that data model. Navigating around the standard to solve their problem helped them gain confidence and quickly grasp the structure of the standard itself. The students could not pretend to become experts in HL7; however, they can surely understand and participate in the design, implementation, and evaluation of information exchange using the HL7 standard.

We have presented an introduction guide to the standard that offers a more effective alternative to a bottom-up approach, learning from first principles. This introduction consists of a guided tour through the complex standard documentation. Presenting the documentation structure to easily navigate it provides enough confidence to search for the needed information. The reading guide and the problems to solve motivated the students to search for answers, enabling thus getting familiar with the documentation easily and rapidly.

We presented the software tools used to help the learning of the syntax, the semantics, and the functions of the HL7 standard, so software development is not needed. The software tools allow to send, receive, and validate messages generated for learning. Specialized XML editors or simple text editors to write HL7 messages are the only needed applications to accomplish the required learning work.

We believe our method for actively learning the HL7 standard can be used by anyone to quickly gain knowledge of this standard.

6.2 Future Scope

HL7 and the Future The adoption of the HL7 standard is allowing for the creation of better tools with which to transfer critical information. This will make every aspect of healthcare more efficient and dynamic and will lessen the opportunity for mistakes.

Perhaps soon, a patient's visit to a clinic or a doctor's office may go like this: A nurse measures a patient's weight, which is electronically populated into his or her medical record. The nurse then takes the patient's blood pressure, and the result is also electronically populated into the medical record. Next, the doctor selects criteria for a lab order, and the lab receives it electronically almost instantly. The lab performs the order and electronically populates the patient's medical record with the results.

HL7 certainly doesn't make these types of technological advancements happen, but the adoption and implementation of the HL7 message standard certainly helps make them possible more quickly.

With the emergence of new HIPAA guidelines and legislation, and as more hospitals and clinics consider the regional health information organization model, EMRs improve, and laboratories and diagnostic imaging centers continue to utilize new technologies into their workflow, HL7 will continue to be a critical component in the evolution of healthcare.

References/Bibliography

---

[1] Fundamentals of Python First Programs by Kenneth A. Lambert[online].Available: https://archive.org/details/Func on Feb. 20, 2019

[2] M. Shell. (2007) IEEEtran webpage on CTAN. [Online]. Available: http://www.ctan.org/tex-archive/macros/latex/ IEEEtran/

[3] Beginning Programming with Python For Dummies by John Paul Mueller.Available: http://www.allitebooks.com/b programming-with-python-for-dummies-2nd-edition Accessed on Feb 20, 2019

[4]Available: https://www.hl7.org.

[5]Available: https://python-hl7.readthedocs.io/en/latest/

[6]Available: https://en.wikipedia.org/wiki/HealthLevel7