

Hands-on Approach for Teaching HL7 version 3

Rita Noumeir and Jean-Francois Pambrun

Abstract—The deployment of the Health Level seven (HL7) version three (v3) standard is steadily increasing. However, the learning curve is long, steep and difficult because the information structure is complex. In this paper, we show how a hands-on approach for teaching HL7 v3 as part of a graduate course on healthcare information, allows students with diversified background and no software development experience, to grasp a large breadth of knowledge in a very short time.

I. INTRODUCTION

Health Level seven (HL7) is a de facto standard for exchanging medical information between systems from different vendors. The standard covers several functional domains such as patient administration, order entry, observation reporting and financial transactions. It has existed for various decades so far and has evolved continuously resulting into newer versions of what is known as version two (v2). Messages contain readable text that is usually communicated over TCP. HL7 knowledge is needed to design and construct information systems in almost all the domains of a healthcare enterprise. It is also needed for deploying and configuring interfaces that are used between systems. In fact, ensuring interoperability with HL7 has required the use of interfaces most of the time. Such interfaces act as bidirectional proxy between communicating peers in order to resolve interoperability problem associated with the use of the standard such as removing some data that the receiving system cannot receive or remapping some data from one place in the message to another place. In parallel to the continuous development of HL7 v2, the development of a completely new version has started more than a decade ago. The new version known as version three (v3) is based on the extensible markup language (XML); it uses the latest development in web technologies such as web services as a transport mechanism and Web Services Description Language (WSDL) for specifying end nodes. HL7 v3 is not widely used yet; however, Canada Health Infoway made it a required pan Canadian standard which will surely help accelerate its deployment. Moreover, HL7 v3 is the standard used in many Integrating the Healthcare Enterprise (IHE) [1, 2] integration profiles which help accelerating not only its integration into products but also its interoperability testing.

The development of the new version took many years. Extensive time has been invested on the data model. One can spend a lot of time learning it before starting to grasp what is possible, i.e. how it can be used to construct a message in order to accomplish an action. This is the approach followed

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and by a grant from Canada Health Infoway.

R. Noumeir and J. Pambrun are with the Department of Electrical Engineering, Ecole de Technologie Supérieure, University of Quebec, 1100 Notre-Dame West, Montreal, Quebec, Canada, H3C 1K3
rita.noumeir@etsmtl.ca

by most experts to learn HL7 v3. It is as a bottom up approach where we learn the details before learning the abstractions or the possibilities; in other words the how before the why or what for. The other way around is to start looking at what can be achieved before examining the details of how to achieve it. This is the approach that we followed in teaching HL7 v3 for a group of graduate students of different backgrounds.

HL7 v3 competence is needed in companies to develop and test healthcare applications that manage or integrate medical information. Hospitals and clinics will need HL7 v3 competence to manage, run and maintain their new generation applications. Government agencies need HL7 v3 competence to provide guidance and specifications for achieving nationwide integration of healthcare information.

HL7 v3 can be more easily understood if one knows or has worked with v2 because both versions cover the same functional domains and semantic message content did not change drastically between versions. However, the learning curve is very long, steep and difficult because the syntax of v3 is new and most importantly the structure of the information differs too much between versions. In v3, the information is modelled and the model is very large and difficult to grasp in its entirety. Moreover, the documentation is done mainly to be browsed. The name of the messages and other artefacts are not easy to remember by a human. Hyperlinks ease the navigation with no doubt, but may cause a novice reader to get lost by turning around. There are many resources available on the web on HL7 v3. The standard text [3] is of course the most complete documentation, but it is a sort of labyrinth. Most of other resources, including the primer book [4], cover the information model along with other general concepts.

In this paper, we present a hands-on method for formally teaching HL7 v3 as part of a graduate course on healthcare information. We also present and discuss the laboratory exercises that have been developed to help the students grasp a large breadth of knowledge in a short time.

II. METHOD

By teaching HL7 v3 as part of a graduate course, we face many challenges: 1- Students have very diversified background; most of them are not necessarily engineers (i.e. one of the students that enrolled for this course was a physiotherapist). 2- Most of the students do not have any experience in software development. 3- Time is limited to seven hours of formal teaching and equivalently for supervised practical assignments.

We start by telling the student where to find a free draft version of the standard [3]. The documentation is very large and very difficult to grasp at first. The novice is quickly lost and may spend a large amount of time navigating the detailed

documentation before getting completely desperate. Our goal is to use the documentation to construct a message quickly, within a three hours period of formal course. Therefore, we have introduced the structure of the documentation in the class and have distributed a step-by-step guide on how to navigate the documentation. This introduction follows a top-down approach, where a specific message is presented first, its structure and finally the data types. We did not cover the entire data model; to the contrary, we covered only the necessary to construct a specific message. The student is expected to be able to come back to the standard for further information. Following the brief formal introduction in class, the students are required to interact with software that is provided to them by constructing specific messages.

We first point out the existence of the chapter entitled Version 3 Guide without getting into its details. This chapter introduces all the concepts used in the version 3 of the HL7 standard. It is important that the students know about its existence so they can get back to it. Our goal is to construct one specific message. Therefore, we start by depicting the overall context where the message lies with respect to the other parts of the standard. We point out the existence of the part Universal Domains that groups many chapters, where each chapter covers one particular functional domain for which messages have been specified. It is important to know the coverage in terms of functional domains, but we choose to work with one domain, the Patient Administration. This is the chapter that covers messages relative to the administration of a patient such as registering a new patient or updating information about an existing one. We present very briefly the Domain Message Information Models (D-MIM) that provide a complete and detailed model of the information used in this domain. D-MIM is so rich that it can be an over killing for a novice. However, it is very important to know where to find the precise definition of concepts and attributes for any information used. The students are informed to revisit this section later for precise concepts definition or relationships between them.

We then browse the interactions and focus on a specific one that is used to register a new patient, the Patient Registry Record Added (PRPA_IN201301UV02) interaction. This interaction is described in a chapter entitled Person Topic; we examine: its structure, since all other domain chapters are organized according to this same structure. After browsing the list of storyboard we focus on the Patient Registry Record Added Story board by noting the name of actors and the interaction. A storyboard depicts the interaction diagram that shows the sequence of interactions between application roles. We then browse the Application Roles and focus more precisely on the roles Patient Registry Informer and Patient Registry Tracker. An application role represents a set of responsibilities to be implemented by an application that sends and/or receives interactions. Next, we browse the Trigger Events and focus on Patient Registry Record Added. A trigger event is a set of conditions that initiate the transfer of information between application roles. After that, we examine the Refined

Message Information Models (R-MIM) and focus on the R-MIM of interest, the Patient Activate. It is a subset of the D-MIM and contains the classes, attributes and associations required to compose the message. Subsequently, we browse the Hierarchical Message Descriptions (HMD) and focus on the HMD of interest Patient Activate. This is the message content. Finally, we browse the list of interactions and focus on the interaction of interest Patient Registry Record Added. Even though we could have used the shortcut directly from the story board, we take the time to learn more about the chapter structure.

The interaction specifies the trigger and the actors. It also specifies the Transmission Wrapper, the Control Act Wrapper and the Message Type that are needed to construct the message. The wrappers are described in the Specification Infrastructure chapter. Before browsing this infrastructure, we overview the HL7 v3 data types described in the Foundation chapter. We browse the data types with an example at hand: a sample message of type Patient Registry Record Added.

Using an XML editor to open the HMD of interest along with the sample message, we study the HMD table view in order to identify optional attributes or sub-structures, multiplicity of attributes, constraints and complex data types, such as code, identifier or address. To understand the semantics associated with each attribute, the R-MIM of interest is revisited. With the XML editor, students start to change the sample message with the help of the schema.

After examining how to construct a message - the payload- we look at how it can be sent. A message needs to be wrapped with a sort of a header that encapsulates information about the sender and receiver (similar to MSH in HL7 version 2) along with some administrative information related to what triggered the interaction (similar to EVN in HL7 version 2). These two structures, the Transmission Act and the Control Act are described in chapters Transmission Infrastructure and Message Control Act Infrastructure, respectively. These specific structures are referenced from the interaction Patient Registry Record Added.

A. Hands-on Laboratory: Interacting with PIXv3 Manager and PDQv3 Supplier

We ask the students to write well formed XML messages using the HL7 standard documentation [3] and the IHE technical framework, more specifically the Patient Identifier Cross-Referencing (PIX) and the Patient Demographics Query (PDQ) integration profiles [2]. We provide server software that fulfils the roles of a PIX Manager and a PDQ Supplier. It can receive messages to register a new patient, to update the demographic information about a registered patient, to respond to queries about demographic and identification information. The server validates the messages received and responds accordingly. Students are required to a new patient, to update patients records and to perform patient cross-reference and patient demographic queries.

In order to perform those tasks students must download and install a PIX&PDQv3 client and server that were de-

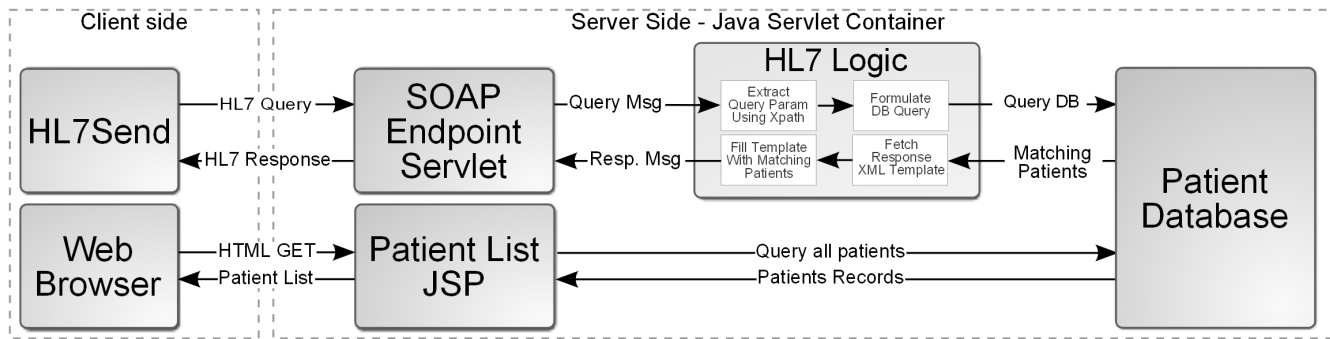


Fig. 1. The architecture of the system.

veloped specifically for this exercise. On the client side, a very simple command line tool was written in JAVA using the Apache Jakarta HttpClient. The client software allows students for example, to send XML query to the PIX&PDQ server and to store the received response; this is achieved with a simple command line interface. For instance, invoking `java jar HL7send.jar -v 3 -f msg.xml -o response.xml -d http://localhost:8080` sends the message `msg.xml` to the `http://localhost:8080` endpoint, and stores the received response in the `response.xml` file.

The server acting as a PIX manager and PDQ supplier is built on JAVA Servlet technology; it is made available to students along the Jetty Servlet container. This makes it easy to install and run as students need only to execute a small batch file to launch the server. The server, originally designed as a testing tool for the IHE PIX&PDQv3 integration profile [5, 6], has been modified for the purpose of this exercise: a database has been added in order to store information and be able to answer HL7 queries by querying the database. The validation capabilities that are used for testing purposes are left unchanged; the messaging framework combined with the newly added database system allows processing of simple HL7 messages. The database comes preloaded with several pre-registered patients. A simple web interface allows students to visualize the full content of the database at any time. Although the exercise described here doesn't rely on the preloaded patients, they can be used to provide more challenging tasks. Both client and server software are solely based on open source software and are freely available on multiple platforms.

B. System Architecture

The architecture of the system that is composed of a server and a client is depicted in Fig. 1.

The system is composed of the following blocks:

1) *HL7Send*: is a utility used by the students to send HL7 messages wrapped in Simple Object Access Protocol (SOAP) envelopes to the server's HL7 endpoint. The returned response is stored in a file.

2) *Web Browser*: is a web browser that can be used by students to view all patients records that are contained in the Patient Database.

3) *SOAP Endpoint Servlet*: is a SOAP Endpoint Servlet that receives HL7 queries from the client. It removes the SOAP envelope and passes the encapsulated message to the HL7 Logic module. The resulting response is then wrapped and sent back to the client. This module makes use of the SOAP with Attachments API for Java (SAAJ) framework.

4) *HL7 Logic*: is an HL7 logic module that parses the incoming messages and performs adequate database queries depending on the request type. It then loads a template response from an XML file and populates it with the data returned from the database. The response message is then transferred to the SOAP Endpoint module to be sent back to the client. The HL7 Logic module uses Xpath to extract query parameters from the incoming message as well as to populate the fields in the response template. Adding support for new messages would involve writing a small function with the adequate Xpath operations as well as writing a response message template. Dom4j is used for the parser and Xpath engine.

5) *Patient List JSP*: is an interface that students can use to view the database content. It is a simple JavaServer Page that queries the database and displays all records in an HTML table.

6) *Patient Database*: is a database that uses the H2 Database Engine which is a full featured SQL database implemented in JAVA. It is embedded in the supplied war file along with the preconfigured patients. This type of database is easy to deploy and does not need any configuration.

This software has been built for education purpose and has some limitations: query continuation and patient merge messages are not supported; demographics query returns only one patient even if multiple records match the request. Also, when the server receives a message that it doesn't know (e.g. patient merge) it will respond with a generic Accept Acknowledgment message but an error message is printed on the server console.

C. Actions executed by students

Using the software, students are required to perform several operations as depicted in the sequence diagram of Fig. 2. The required operations consist in:

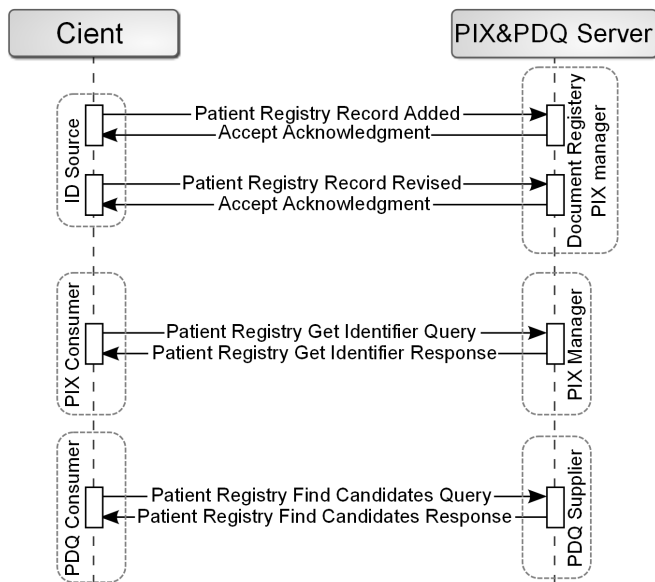


Fig. 2. Messages constructed by students.

- Registering one patient in domain A; this is achieved by issuing a Patient Registry Recoded Added message.
- Registering the same patient in domain B, but with a misspelled name; this is achieved by issuing a Patient Registry Recoded Added message.
- Correcting the misspelled name by updating the record with the correct information; this is achieved by issuing a Patient Registry Recoded Revised message.
- Querying the PIX manager with the ID from domain A to retrieve the ID for the same patient in domain B; this is achieved by issuing Patient Registry Get Identifier Query message.
- Querying the PDQ supplier using enough demographics to get the patient registered in domain A in the response; this is achieved by issuing a Patient Registry Find Candidate Query message.

HL7 messages are encapsulated in SOAP envelopes that specify the action being performed and its intended destination [7]. Students are requested to properly construct the SOAP wrapper as well as the encapsulated message. Important fields of the SOAP wrapper include target destination, reply destination and the intended action. Failure to properly set these attributes will result in an unsupported message error.

III. RESULTS

Ten students with diversified background enrolled for the course and undertook the work. Their background included electrical, telecommunication, mechanical and computer engineering, as well as clinical background with no engineering or software development knowledge. For the practical realization of the exercises, one person who knows the software and with experience in the subject matter provided supervision during two periods of three hours each. The students were extremely

motivated. They all succeeded in accomplishing the requested work.

As part of their evaluation, students were requested to provide the messages they constructed for review by the instructor and to comment on the received responses. The learned skills include navigating the HL7 and IHE documentation, constructing HL7 messages, writing XML messages according to schemas, understanding acknowledgements and responses. Students also get experience with typical HL7 interaction workflows and understand the roles of IHE PIX manager, PIX consumer, PDQ supplier, PDQ consumer and ID source.

IV. CONCLUSIONS

Learning HL7 can be achieved with a hands-on approach. Real problems motivated the students to search for the answers within the standard documentation enabling them to become familiar quickly with it. The students did not use any book on the subject. However, we have provided them with a small guide on how to navigate through the standard documentation; a sort of a guided hopping around in the documentation that proved to be sufficient. In fact, the standard documentation is the only complete resource. Showing the students how to navigate around the documentation gave them enough confidence to dive into the documentation details.

REFERENCES

- [1] Noumeir R., Integrating the Healthcare Enterprise Process, *Int. J. Healthcare Technology & Management*, v 9, n 2, 167-180, 2008
- [2] IHE Technical Framework and supplements, [Online] Available http://www.ihe.net/Technical_Framework/index.cfm, June 2, 2010
- [3] HL7 Version 3 Ballot Web Site Archive, [Online], <http://www.hl7.org/v3ballot/html/index.htm>, May 18, 2010
- [4] Hinchley A, Understanding Version 3: A primer on the HL7 Version 3 Healthcare Interoperability Standard Normative Edition, Alexander Monch Publishing, 2007.
- [5] Testing tools source code of the Integrating the Healthcare Enterprise (IHE) PIX and PDQ v3 Integration Profiles, [Online] Available <http://sourceforge.net/projects/ihe-pixpdqv3>, June 5, 2010
- [6] PIX/PDQ V3 Software, [Online] Available <http://ihe.etsmtl.ca>, June 5, 2010
- [7] Tutorials on SOAP, HTTP, WSDL, [Online], <http://www.w3schools.com>, April 20, 2010