Teaching DICOM by Problem Solving

Rita Noumeir · Jean-François Pambrun

Published online: 5 April 2012
© Society for Imaging Informatics in Medicine 2012

Abstract The Digital Imaging and Communications in Medicine (DICOM) is the standard for encoding and communicating medical imaging information. It is used in radiology as well as in many other imaging domains such as ophthalmology, dentistry, and pathology. DICOM information objects are used to encode medical images or information about the images. Their usage outside of the imaging department is increasing, especially with the sharing of medical images within Electronic Health Record systems. However, learning DICOM is long and difficult because it defines and uses many specific abstract concepts that relate to each other. In this paper, we present an approach, based on problem solving, for teaching DICOM as part of a graduate course on healthcare information. The proposed approach allows students with diversified background and no software development experience to grasp a large breadth of knowledge in a very short time.

Keywords Digital Imaging and Communications in Medicine · DICOM · Medical imaging · Healthcare information · Teaching · Problem solving

Introduction

The Digital Imaging and Communications in Medicine (DICOM) [1] is the de facto standard for encoding and communicating medical images and imaging-related information. DICOM has existed for many decades so far. It is

widely used within the radiology department for encoding images from all modalities including X-ray, magnetic resonance (MR), computed tomography (CT), and ultrasound. It is also used to communicate medical images between different systems that may be provided by different manufacturers. DICOM has evolved to permit the encoding of other imaging information such as imaging reports, key object selection, and presentation states; it is also used outside of the radiology department for encoding ophthalmology, dentistry, and pathology images. Furthermore, it is used to manage cooperated workflows between various systems by enabling the use of worklists and notification events; recently, it has defined standard mechanisms based on web technologies and web services to enable access to images and imaging information from outside the imaging department.

Learning and mastering DICOM is needed by any engineer or an Information Technology (IT) professional who aims at working in the domain of medical imaging. So far, learning DICOM has been achieved outside the academic environment, probably either because universities have considered it a practical field that lies outside of their mission or because of the lack of experience and interest in DICOM from the academic people. We believe that this is changing, mainly due to the excessive demand in highly skilled personnel trained in technologies related to medical imaging. This demand comes from many sectors including industrial, hospitals, and governmental agencies.

The need for such competence by industrial, clinical, and governmental sectors is exploding, driven mainly by the increasing demand for interoperability in healthcare, which is enabled by an increasing number of projects for integrating many facilities or for deploying regional or nationwide Electronic Health Records (EHR). EHR is a distributed system that allows access to diagnostic information for assisting in health decisions, regardless of the geographic

R. Noumeir (✉) · J.-F. Pambrun
Department of Electrical Engineering,
École de Technologie Supérieure,
1100 Notre-Dame West,
Montreal, QC H3C 1K3, Canada
e-mail: rita.noumeir@etsmtl.ca

location of the point of access or the institution where the information was initially gathered. Because EHR requires the cooperation of several heterogeneous distributed systems, interoperability is essential. Achieving interoperability requires the use of standards, such as DICOM, for exchanging medical images or imaging-related information. Because DICOM has been in use for many decades so far within the radiology domain, medical images are among the first diagnostic information to be shared between institutions and between various specialists. This is the case in Canada where projects for sharing images between multiple hospitals are already being deployed. DICOM enabled the shift to filmless radiology. Therefore, images have been used in a digital format for some time so far; this made radiology the pioneer that paved the road to EHR implementations, enabling distributed healthcare systems. It is apparent with the Integrating the Healthcare Enterprise (IHE) initiative whose goal is to provide a framework for achieving interoperability in healthcare [2]. IHE started in 1998 and was initially jointly sponsored by the Radiological Society of North America and the Healthcare Information and Management Systems Society. The first IHE integration profile, the Radiology Scheduled Workflow, provides a solution for integrating multiple systems involved in the fulfillment of a radiology order [3]. Currently, several other associations sponsor IHE that has expanded over several clinical domains and who benefits from broad international support. IHE activities are solving integration problems in many clinical domains, such as radiology, laboratory services, cardiology, radiation oncology, ophthalmology, and healthcare devices. Moreover, IHE is setting up the foundation for the EHR by enabling interoperability among care domains within a single healthcare enterprise and across many healthcare enterprises. For example, the Cross-Enterprise Document Sharing for imaging Integration Profile (XDS-I) enables the sharing of imaging information such as images and imaging diagnostic report across health enterprises. XDS-I uses DICOM for encoding the document that is shared; it also permits DICOM as the communication protocol for retrieving images [4].

Companies need DICOM competence to develop and test healthcare applications that manage or integrate medical images. Hospitals and clinics need the DICOM competence to manage, run, and maintain their installation base of IT applications that deal with medical images. Government agencies need DICOM competence to provide guidance and specifications for achieving nationwide integration of healthcare information.

We are not aware of an academic curriculum that includes teaching the DICOM standard. This gap has been covered by specific trainings provided, either by consultants, or by training firms specialized in seminars that usually cover a general introduction to the subject. On the other hand, a very small number of books exist [5–8]. These books are

intended mainly for engineers because they focus on the engineering aspects of the standard. They provide an introduction to DICOM and usually focus on one aspect more than the others such as encoding binary data [8], exchanging data over the network [5], or encoding structured reports [6].

Teaching DICOM as part of a graduate course presents 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; and (3) supervision time is limited to a few hours.

In this paper, we present the method that we followed to formally teach DICOM as part of a graduate course on healthcare information. We also present the hands-on experiments that have been developed to help students grasp a large breadth of knowledge in a short time.

Method

We start with a formal course of 3 h during which we present the structure of the DICOM documentation along with its basic concepts. We guide the students through the documentation that is very large and hard to read; like any standard documentation, it cannot be read in a sequential linear way. In class, we always refer to the documentation while presenting the concepts. We do not provide the student with any document that explains the standard. However, we provide them with a guide on how to non-sequentially read the documentation. For every concept explained, we provide a reference to the documentation that helps its understanding. The references are usually multiple and may be explained in different parts of the documentation. They illustrate the definition, the relation to other concepts, or point to concrete examples.

Many abstract concepts are defined by DICOM, i.e., a Service Object Pair (SOP). These specific concepts constitute a basis that needs to be mastered in order to work with the standard; when they are used by the students, they become more concrete. Our goal is to provide the students, very quickly, enough information about those concepts. In this way, they are able to solve real problems, such as understanding or debugging a communication between peers, or constructing a DICOM file from non-DICOM information. In the next subsections, we first provide a brief description of the guide to read the standard documentation and then present the hands-on experiments used to help grasping DICOM concepts.

How to Read the DICOM Standard Documentation

We start by telling the students where to find the standard documentation so they can download it. To put the

standard's benefits in context, we depict a common radiology workflow: order entry at the radiology information system, scheduling of the imaging procedure, image acquisition at the modality equipment, image storage at the image archiving system, diagnostic report generation at the image visualization workstation, and report delivery to a clinical workstation or a hospital information system. For every step in this workflow, we describe at a very high level how DICOM enables its implementation between heterogeneous systems.

Then, we explain the structure of the documentation by describing the content of the various chapters: the encoding of information covered in chapters 3, 5, and 6; the media exchange of information covered in chapters 10, 11, and 12; the encoding of structured information covered in chapters 3 and 16; the HTTP communication covered by Web Access to DICOM Persistent Objects (WADO) in chapter 18; and the communication covered in chapters 4, 7, and 8. We point out that the structures and the chapters' content are described in volume 1 of the standard, in sections 5 and 6.

To help students work with different documents of the standard, at different development stage, we describe the standard development process: working groups composed mainly of experts from the industry, each specialized in a specific area such as visualization; process for augmenting the standard resulting in the publication of supplements whose content is bundled into the next standard release; correction process resulting in publishing of correction proposals whose contents are also bundled in the standard's next release. Getting familiar with these development processes enables the students to navigate through the various versions of the standard, especially that all these documents are freely available on the web.

Information Encoding

DICOM information is encoded in a binary format. Therefore, to view the content of a DICOM file, a special software is needed to “dump” or transform its content into a readable text output. We provide the students with a sample file whose content is dumped into a text format. We present the binary encoding such as little endian or big endian and browse with the students the various data types, or value representations.

DICOM has a basic structure where each element of information is uniquely identified with a tag composed of a group number and an element number; this structure is presented to students by looking at examples from the dictionary, such as Patient's Name or Echo Time. Figure 1 shows a binary fragment that represents two data elements of a DICOM file.

Each element of information has a specific type; some may have multiple values. Therefore, we describe the

organization of the DICOM dictionary that lists all possible tags along with their data type and their possible repetition. DICOM allows the explicit encoding of the value representation within the file, in which case the data type of each information element is encoded. In an implicit encoding, the data type is not encoded and a dictionary is needed to decode the file. Therefore, we explain the transfer syntax by presenting the differences between big endian or little endian, and between implicit or explicit value representations. We also present the presence requirements of attributes that can be mandatory and not empty (type 1), optional (type 3), mandatory eventually empty (type 2), or conditional (type 1C or type 2C). DICOM also allows the grouping of attributes into sequences and the imbrications of sequences. To illustrate the use of sequences, we present the Patient Module and point out the sequences of identifications.

Data Model

DICOM uses a data model known as the Information Object Definition (IOD). IOD instances are arranged hierarchically: instances are grouped into Series, and Series are grouped into Studies; each instance, series, or study is identified by a unique identifier (UID); every institution that generates IODs assigns a UID to each generated DICOM file. We provide the students with several files and analyze with them the hierarchical structure of the IODs encoded in those files. We also inform them about the procedure to obtain a unique prefix and describe how the generation of unique IDs is usually achieved.

IOD Definition

Each image type such as an MR or CT is encoded as a specific IOD. The information elements that compose an IOD are grouped into Information Entity (IE), such as Patient or Study IE. This is well summarized in [5]. This composition is important because it depicts, at a high level, the various parts that make an image for example. Therefore, we browse with the students the list of the IEs that compose each image type (by analyzing part 3, Table A1-1). Furthermore, we look into the details of some specific IEs, such as the Patient Demographics, the General Series, and the Image Pixel Modules.

Services or Operations

We present to the students, very briefly, the operations that can be performed on an IOD. DICOM defines an operation in terms of a service such as storing an image, querying for images that belong to a specific patient, or querying for a list of work items to do. Services are defined in part 4. Examples of services include Verification, Storage, Query/

tag	type	len	data	tag	type	len	data	...
...10 00 10 00 50 4E 10 00 4A 6F ...				10 00 30 00 44 41 08 00 32 30 30 37 30 38 32 32 ...				
(0010,0010)	PN	16	Jo..	(0010,0030)	DA	8	2007-08-22	

Fig. 1 A binary fragment from a DICOM file showing two data elements: 4 bytes represent the patient's name tag; 2 bytes represent the data type *PN* (person's name); 2 bytes contain the length of the

data; 16 bytes contain the patient name; 4 bytes represent the patient's date of birth tag; 2 bytes represent the data type *DA* (date); 2 bytes contain the length of the data; 8 bytes contain the patient's date of birth

Retrieve, Basic Worklist Management, Print Management, Storage Commitment, and Modality Performed Procedure Step. The pair composed of a service and an IOD is called a SOP. DICOM identifies each SOP type with a UID called a SOP class. We browse with the students the list of all SOP classes that are provided in part 6 of the standard. Furthermore, we examine the details of a specific SOP class, the one that identifies an MR image.

In order to understand an operation that takes place between two application peers, we present the concept of an Application Entity (AE), and describe the role that AE can play during an interaction: the user role played by an AE which invokes operations and the provider role played by an AE which performs operations. We also browse, very quickly, part 4 by looking at the Verification, Storage, Study root Query/Retrieve, and Modality Query services.

Communication

We present very briefly the communication layers that DICOM defines over the Transmission Control Protocol (TCP). We also present very briefly how a communication is established and how services and transfer syntaxes are negotiated and accepted.

Construction of DICOM Files

We ask the students to generate three different types of DICOM files: (1) a DICOM secondary capture that wraps a specific image; (2) a DICOM structured report [9] that contains a specific text within a specific section structure; and (3) a Key Object Selection (KOS) that references a specific DICOM image. We also ask the students to validate the DICOM files they generate and to visualize them.

In order to solve this problem, we ask the students to download and install an open source implementation of DICOM in Java, the Dcm4che2 [10] software. We also ask them to download a DICOM validator, the dciodvfy [11], as well as a DICOM viewer, the open source software ezDicom [12], or DICOMscope [13] software. In order to make it easier for students with very limited knowledge in software development, we provide basic scripts that prepare the necessary environment for running the downloaded software. The students generate the DICOM file using tools provided by Dcm4che2. These tools (txt2dcm, jpg2dcm, and xml2dcm) take a text or XML input file that contains the DICOM tags

along their corresponding values and generates the DICOM output file, as shown in Fig. 2. The students only need to provide the adequate input file; they are thus relieved from dealing with the binary aspects of the encoding.

The generation of DICOM files presents challenges of increasing difficulties. Generating a DICOM image is fundamental; it allows students to learn all the basic DICOM structure and encoding at the syntactic level as well as the semantic one. The students need to identify all the necessary information modules needed to encode a secondary capture SOP class. They also need to generate valid study, series, and instance unique identifiers. Moreover, they need to identify all the required information with their encoding constraints and provide semantically reasonable information to populate these fields. Figure 3 shows information provided by students to generate the image whose header is depicted in Fig. 4. The learned skills include the DICOM data model and syntax, as well as parts 3, 5, 6, and 10 of the standard documentation.

On the other hand, generating a DICOM structured report presents additional challenges. The students are provided with a DICOM image and the text of a radiology report. They are required to generate a DICOM report for the specified image. Therefore, the students have to reuse the same study and patient information provided in the image. They also have to provide a minimal structure for the report's content. The developed skills include a deeper understanding of the DICOM data model and its semantics. They learn about encoding sequences. Generating a report that belongs to the same study as a specific image helps

Fig. 2 Students have to provide a valid file (.cfg) containing all the required tags and invoke jpg2dcm in order to create a secondary capture image

Exercise1.cfg

```
# Patient Module
00100010:Smith^John
00100020:1365457
00100030:19840824
00100040:M
(...)
# General Study Module
0020000D:2.16.124.32164
00080020:20110615
00080030:131646.0
00080090:Lisook^Steven
00200010:1784
00080050:THU9948
00081030:CHEST/ABD
(...)
# Serie Module
0020000E:2.16.124.159753
(...)
```

Fig. 3 Excerpt from the information provided by students (.cfg file) to produce the image file depicted in Fig. 4. The format of each line is tag: value, where tag is composed of the group and element numbers. The second line, for instance, represents the patient's name whose tag is (0010, 0010)

them learn about how instances are related and grouped into series and studies, and requires them to identify the levels of information. They also learn about how to read part 16 of the standard where constraints to structures are described.

Generating a DICOM KOS represents even more challenges. The students are provided with a DICOM image and a sample KOS file. They are required to generate a DICOM KOS that references the given image. They can use the dcm2xml tool that comes with Dcm4che2 in order to reverse engineer the provided KOS, as shown in Fig. 5. The generated KOS has to belong to the same study as the image; therefore, the complete study level information needs to be shared between the image and the KOS. Moreover, the KOS header has to include a sequence that contains all instances that are referenced inside the KOS content; this represents a higher degree of complexity. Furthermore, the KOS content needs to follow the structure specified by the template ID 2010, described in part 16 of the standard documentation. Therefore, students acquire a deeper knowledge of DICOM sequence encodings and templates.

Analysis of DICOM Communication

DICOM specifies communication handshakes between peers on top of TCP. Communication takes place in binary format and involves negotiations of services that are needed or can be provided by peers, the roles applications play such as the role of user or provider as well as transfer syntaxes that can be used and the one that is chosen. The most used services in practice include C-ECHO that is used to “ping” another application using full negotiation and hand shaking, C-STORE to send an image or information instance to a peer, C-FIND to query for instances that match specific search parameters, and C-MOVE to request a specific instance.

In order to introduce students to DICOM communication, we require that they sniff and analyze a complete set of interactions between two peers, using a network analyzer. The open-source Wireshark network protocol analyzer [14] is used to perform the network capture and analysis. Wireshark comes preloaded with modules that are capable of presenting several network protocols in a human readable format. Therefore, it has the ability to translate the binary format of DICOM data into a plain text that is readable by a human: it displays the content of the messages along with the DICOM tags of the data being transferred.

Configuring and using Wireshark is not difficult; students start the network traffic analyzer to capture and analyze traffic of interest by entering “dicom” in the filter field. Students are asked to use dcm4che2 toolkit to run a service provider and to require several DICOM operations by running a service user. They are also required to analyze the network traffic between the server provider and the service user using Wireshark. For instance, students start a C-STORE SCP by invoking “dcmrcv < EATitle>:<port> -dest < file destination>”; they send a C-ECHO with “dcmecho < AETitle > @ < ip>:<port>”; they require to store an image using “dcmsnd < AETitle > @ < ip>:<port> file.dcm”. With Wireshark, students listen to the network communication and learn about the DICOM handshake by analyzing the details.

We take the difficulties a bit further by providing the students with a file containing traffic details that we have captured in advance, without giving them any detail about the operation that was performed. We ask them to analyze and describe every DICOM packet present in the capture. We also ask them to find the goal of each message, the patients involved, the modalities queried for, and the use case performed. Table 1 shows a fragment of the analyzer output. To carry out this work, students need to find their ways through the DICOM documentation and to understand the communication handshakes and details.

We also provide the students a file containing traffic details of a failed communication that we have conducted in advance, without giving them any details about neither

Fig. 4 Some common tags are displayed in a tabular format, where shaded cells need to be provided by students, in either xml or text (.cfg) format

Tag	Tag Meaning	VR	Data
...	...	...	...
(0002,0010)	Transfer Syntax UID	UI	1.2.840.10008.1.2.4.50
...	...	...	...
(0008,0016)	SOPClassUID	UI	1.2.840.10008.5.1.4.1.1.7
(0008,0018)	SOPInstanceUID	UI	2.16.124.32164
(0008,0020)	StudyDate	DA	20110615
...	...	...	...
(0010,0010)	PatientName	PN	Smith^John
(0010,0020)	PatientID	LO	1365457
(0010,0030)	PatientBirthDate	DA	19840824
(0010,0040)	PatientSex	CS	M
...	...	...	...
(0020,000D)	StudyInstanceUID	UI	2.16.124.963248
(0020,000E)	SeriesInstanceUID	UI	2.16.124.159753
...	...	...	...
(0028,0010)	Rows	US	512
(0028,0011)	Cols	US	512
(0028,0100)	Bits Allocated	US	8
(0028,0101)	Bits Stored	US	8
...	...	...	...
(7FE0,0010)	Pixel Data	OB	

the operation that was performed nor the eventual problem. The problem that we induce is a communication rejection by the receiving peer because the application entity of the peer that is issuing the communication is not recognized. In this case, the association is rejected with a listed reason of "called-AE-title-not-recognized." Students need to examine the binary content alongside the DICOM documentation to

interpret the issue. This experience not only provides the students with the motivation to dive into the documentation but also exposes them to communication problems commonly encountered in practice and provides them with the tools to diagnose and understand the communication issues.

Results

We used the work described in this paper as part of a graduate course that has been offered to students with major in healthcare or information technology. Two groups of ten students each enrolled for the course and completed the work. The students had diversified backgrounds that 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 experiments, one person with experience in the subject matter provided supervision during two periods of 3 h each. The students were extremely motivated. They all succeeded in accomplishing the requested work.

Students were required to provide the DICOM files they generated: the secondary captured image, the structured

Fig. 5 Students are asked to analyze a complex DICOM document, such as a key object selection or a structured report, using the dcm2xml tool. They can then modify the exported xml file and re-transform it into a DICOM document using xml2dcm

Table 1 The network analyzer output: each line contains the packet number (only DICOM interactions are displayed here), the elapsed time counted from the moment the capture is started, the source IP, the destination IP, the protocol, the packet length, and information about the packet content

No.	Time	Source	Destination	Protocol	Length	Info
4	0.050864	10.196.113.104	10.196.113.131	DICOM	257	A-ASSOCIATE request DCMECHO → DCMSRV
7	0.249749	10.196.113.131	10.196.113.104	DICOM	174	A-ASSOCIATE accept DCMECHO ← DCMSRV
8	0.298591	10.196.113.104	10.196.113.131	DICOM	134	P-DATA, C-ECHO-RQ ID=1
9	0.302479	10.196.113.131	10.196.113.104	DICOM	144	P-DATA, C-ECHO-RSP ID=1
10	0.319905	10.196.113.104	10.196.113.131	DICOM	64	A-RELEASE request
11	0.320771	10.196.113.131	10.196.113.104	DICOM	64	A-RELEASE response
13	0.32805	10.196.113.131	10.196.113.104	DICOM	64	A-RELEASE request
19	146.421726	10.196.113.104	10.196.113.131	DICOM	292	A-ASSOCIATE request DCMSND → DCMSRV
22	146.614104	10.196.113.131	10.196.113.104	DICOM	174	A-ASSOCIATE accept DCMSND ← DCMSRV
24	146.63303	10.196.113.104	10.196.113.131	DICOM	132	P-DATA, C-STORE-RQ ID=1, Basic Text SR Storage
26	146.698619	10.196.113.131	10.196.113.104	DICOM	220	P-DATA, C-STORE-RSP ID=1
27	146.715863	10.196.113.104	10.196.113.131	DICOM	64	A-RELEASE request
28	146.716419	10.196.113.131	10.196.113.104	DICOM	64	A-RELEASE response
31	146.733554	10.196.113.131	10.196.113.104	DICOM	64	A-RELEASE request
36	170.911462	10.196.113.104	10.196.113.131	DICOM	328	A-ASSOCIATE request DCMSND → DCMSRV
39	171.104534	10.196.113.131	10.196.113.104	DICOM	208	A-ASSOCIATE accept DCMSND ← DCMSRV
54	171.124073	10.196.113.104	10.196.113.131	DICOM	1,514	P-DATA, Secondary Capture Image Storage Fragment
69	171.124421	10.196.113.104	10.196.113.131	DICOM	714	P-DATA, Secondary Capture Image Storage Fragment
82	171.125747	10.196.113.104	10.196.113.131	DICOM	132	P-DATA
93	171.478124	10.196.113.131	10.196.113.104	DICOM	216	P-DATA, C-STORE-RSP ID=1
94	171.489027	10.196.113.104	10.196.113.131	DICOM	64	A-RELEASE request
95	171.490729	10.196.113.131	10.196.113.104	DICOM	64	A-RELEASE response
98	171.495631	10.196.113.131	10.196.113.104	DICOM	64	A-RELEASE request
103	254.878258	10.196.113.104	10.196.113.131	DICOM	567	A-ASSOCIATE request DCMQR → DCMSRV
106	255.075323	10.196.113.131	10.196.113.104	DICOM	232	A-ASSOCIATE accept DCMQR ← DCMSRV
107	255.125117	10.196.113.104	10.196.113.131	DICOM	230	P-DATA, C-FIND-RQ ID=1, C-FIND-RQ-DATA
108	255.146517	10.196.113.131	10.196.113.104	DICOM	154	P-DATA, C-FIND-RSP ID=1
110	255.345362	10.196.113.131	10.196.113.104	DICOM	978	P-DATA, C-FIND-RSP ID=1
111	255.362676	10.196.113.104	10.196.113.131	DICOM	64	A-RELEASE request
112	255.363189	10.196.113.131	10.196.113.104	DICOM	64	A-RELEASE response
115	255.379129	10.196.113.131	10.196.113.104	DICOM	64	A-RELEASE request

Four DICOM interactions are shown in this table: a C-ECHO depicted in lines 4 to 13; a C-STORE of a text report depicted in lines 19 to 31; a C-STORE of a secondary image capture depicted in lines 36 to 98; and a C-FIND query depicted in lines 103 to 115

report that included a reference to an image, and the key object selection that referenced a specified image. They were also required to provide the result of their analysis of a DICOM communication handshake, as well as to guess the reason of the communication failure provided to them by pointing to communication excerpts as evidences that support their answers.

By successfully performing the work required, the students developed the following skills:

- Understanding the functional areas covered by DICOM such as encoding of images, reports, and key object selection; media exchange; network communication; and web communication.
- Understanding the documentation structure in order to be able to look and find information in the documentation quickly.
- Understanding the DICOM model of the real world.
- Becoming familiar with DICOM communication layers and able to interpret information exchange between peers.
- Becoming able to construct a DICOM information object.
- Becoming familiar with imaging workflow and the relationships between order, image, report, and other imaging information objects.

The problems on which the students worked are inspired from real clinical scenarios. The creation of a DICOM image is the most basic step in a clinical imaging process.

The creation of a report from an image is the “raison d’être” of the imaging process. The report is the piece of information that is shared with referring physicians and with professionals outside of the imaging department. Working with a simple structured report introduced the students to the domain of information structuring and to codes. Structure and codes are the basic blocs of semantics that are at the core of next-generation information systems [15]. The creation of a manifest (or KOS) is important to the EHR. In fact, the deployment of XDS-I [4] as the framework for sharing images within regional and national EHRs is taking place in many countries including Canada, USA, Japan, and several European countries. EHR is expected to improve the quality of care by enabling access to all patients’ relevant information to assist in health decisions. It is also expected to improve the efficacy and efficiency of healthcare by improving productivity and by reducing the duplication of tests.

The skills learned from solving the practical problems presented here are needed in the context of sharing images within the EHR. In fact, the XDS-I architectural model is based on a central registry that holds metadata describing every published document, and responding to queries about documents meeting specific criteria. The registry maintains information about the location from which documents may be retrieved, making possible the deployment of one or multiple document repositories that persistently store the documents and respond to document retrieval requests. In order to share a set of images, a DICOM manifest that contains references to a set of DICOM instances is published. With this solution, the manifest is published and not the images. When a system needs to retrieve the images, it retrieves the manifest, decodes it to get the list of referenced images, and issues a DICOM transaction, such as retrieve (C-MOVE) or WADO, to retrieve the images from the source system. With this architecture, the DICOM manifest is the published document, and the consumer system communicates with the source system using either a DICOM web transaction or a common DICOM C-MOVE transaction. Therefore, the skills acquired from the proposed work enable the students to be very familiar with the EHR deployment in Canada and other countries.

To supplement the skills acquired from the proposed work, students were also requested to read and learn about a specific subject, not covered in the class, and to present their subject to their peers, during a period of 30 min. Student were provided with a list of subjects covering several DICOM aspects that included DICOM conformance, removable media, security, compression, gray scale consistency, printing, hanging protocols, advanced information objects such as enhanced CT, and workflow services such as worklists. The presentations performed by the students complemented the curriculum offered in the class and

proved that all of them were able to study, present, and discuss in detail a new subject from the DICOM standard.

Conclusion

We have shown that learning DICOM can be achieved very quickly with a problem solving approach. DICOM is a practical standard. Real problems motivated the students to search for the answers within the standard documentation enabling them to become familiar quickly with it. Our method, implemented as part of a graduate course, succeeded in overcoming many challenges: (1) students with diversified background where most of them are not necessarily engineers all succeeded in becoming familiar with the DICOM data model, encoding, and communication; (2) students with no experience in software development were able to solve the practical problems they faced by using and combining open-source software; and (3) students were able to achieve the work in a time limited to 10 h of formal teaching and equivalently for supervised practical assignments. This was possible because, with real problems to solve, the students were guided to search in the standard documentation in an effective way. Browsing the documentation to solve a problem motivated and accelerated the learning. We believe that our method can help any person who needs to learn DICOM to become familiar with this standard very quickly.

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. Because the standard documentation is the only complete resource that is also freely available, showing the students how to navigate around it gave them enough confidence to dive into the details.

The students were not required to write software. Therefore, software development skills were not prerequisite to learning the DICOM standard. However, students used open-source software to solve the various practical problems that they faced. The various software tools they used can be very useful for them later in their work environment, whether it is the industry, the hospital, or a governmental agency. The open-source network protocol analyzer can be used to identify interoperability problem between two DICOM communicating peers, such as missing or wrong communicated information, incompatible transfer syntaxes, or incompatible communication behaviors. The open-source DICOM library can be used to dump and analyze the content of a DICOM file in order to identify the content of a specific data field or to identify missing information that would be necessary for ensuring interoperability. The open-source image, report, KOS, and presentation states viewer can be used to

compare the display of an image or an information object with third-party viewers in order to verify that information is rendered correctly. Analyzing file's content and displaying it enabled the students to identify incompatible or misuse of information that would need specific attention in order to ensure interoperability and consistency of display.

Acknowledgments This work was supported by a grant from École de Technologie Supérieure and the Natural Sciences and Engineering Research Council of Canada.

References

1. The Digital Imaging and Communications in Medicine (DICOM) standard, (2011, October 22), [Online], <http://medical.nema.org>
2. Noumeir R: Integrating the healthcare enterprise process. *Int J Healthc Technol Manag* 9:167–180, 2008
3. IHE Technical Framework and supplements, (2011, October 22), [Online], http://www.ihe.net/Technical_Framework/index.cfm
4. Noumeir R., Pambrun J.F.: Images within the electronic health record. *IEEE Int Conf Image Process ICIP09* 1761–1764, 2009
5. Revet, B., DICOM Cook Book for Implementations in Modalities, Philips Medical Systems, 1997, (2011, October 22), [Online], <http://www.archive.org/details/DicomCookbook>.
6. Clunie, D. A., DICOM Structured Reporting; PixelMed Publishing, Bangor, PA, 2001, (2011, October 22) [Online], <http://www.pixelmed.com/>
7. Oosterwijk H, Gihring PT: DICOM Basics, 3rd edition. OTech, Inc., Aubrey, 2002
8. Pianykh OS: Digital Imaging and Communications in Medicine (DICOM)—A Practical Introduction and Survival Guide. Springer-Verlag Berlin Heidelberg 2012
9. Noumeir R: DICOM structured report document type definition. *IEEE Trans Technol Biomed* 7(4):318–328, 2003
10. Dcm4che2 DICOM Implementation in JAVA, (2011, October 22), [Online], <http://sourceforge.net/projects/dcm4che/>
11. DICOM Validator - dciodvfy, (2012, January 4), [Online], <http://www.dclunie.com/dicom3tools/dciodvfy.html>
12. ezDicom Dicom Viewer, (2011, October 22), [Online], <http://sourceforge.net/projects/ezdicom/>
13. DICOMscope Dicom Viewer, (2011, October 22), [Online], <http://dicom.offis.de/dscope.php.en>
14. Wireshark network protocol analyzer, (2011, October 22), [Online], <http://www.wireshark.org/>
15. Noumeir R: Benefits of the DICOM structured report. *J Digit Imaging* 19(4):295–306, 2006