Computer Standards & Interfaces xxx (xxxx) xxx–xxx

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

A community-driven validation service for standard medical imaging objects

Jorge Miguel Silva*, Tiago Marques Godinho, David Silva, Carlos Costa

DETI/IEETA, University of Aveiro, Portugal

ARTICLE INFO

Keywords:
Web
DICOM
PACS
Medical imaging
Validator

ABSTRACT

Digital medical imaging laboratories contain many distinct types of equipment provided by different manufacturers. Interoperability is a critical issue and the DICOM protocol is a de facto standard in those environments. However, manufacturers' implementation of the standard may have non-conformities at several levels, which will hinder systems' integration. Moreover, medical staff may be responsible for data inconsistencies when entering data. Those situations severely affect the quality of healthcare services since they can disrupt system operations. The existence of software able to confirm data quality and compliance with the DICOM standard is important for programmers, IT staff and healthcare technicians. Although there are a few solutions that try to accomplish this goal, they are unable to deal with certain situations that require user input. Furthermore, these cases usually require the setup of a working environment, which makes the sharing of validation information more difficult. This article proposes and describes the development of a Web DICOM validation service for the community. This solution requires no configuration by the user, promotes validation results' share-ability in the community and preserves patient data privacy since files are de-identified on the client side.

1. Introduction

In recent decades, healthcare institutions have been continuously increasing the production of digital medical imaging data. In part, this was due the increase of digital medical imaging equipment and information systems, which are now fundamental in medical diagnosis, decision support and treatment procedures. Picture Archiving and Communication System (PACS) is predominant in this field, providing tools for data acquisition, storage, distribution and visualization. It is a mature concept supported by a set of hardware and software technologies, being grounded in the Digital Imaging and Communications in Medicine (DICOM) standard to ensure normalized data formats and processes. It is a universally accepted standard in medical imaging laboratories, designed to encompass all functional aspects [1–3]. Nowadays, the communication between equipment and information systems is usually done using the DICOM standard [4]. This defines the reference information model, how data is encoded and communicated. Data is merged in structured objects that follow normalized templates per image modality, which contain metadata related with the procedure, patient, acquisition technique and institution, besides pixel data.

Regular workflows are so supported by PACS [5,6], that the existence of non-conforming applications or equipment may disrupt the regular operation with potential losses in the medical undertaking [7].

Despite the existence of DICOM standard, the reality is that challenges to interoperability still arise. Furthermore, technology is constantly evolving and DICOM needs to be updated, thus, hindering compliance between equipment.

The baseline to ensure interoperability between different systems is the DICOM Conformance Statement, since it provides a foundation to determine connectivity and assess the potential interoperability of two products. In some cases, it is possible to identify potential problems without ever having the products physically connected. It is a public document that must be provided by the vendor which describes the DICOM capabilities and functions implemented in a product, allowing connectivity comparisons and defining all the necessary information to perform a certain functionality [8]. DICOM validation software is important to assist in the testing of products' DICOM conformance, providing an independent measurement of the accuracy of products' DICOM interface.

Notwithstanding, verifying the compliance of data produced by PACS applications is not trivial, since the DICOM standard supports a significant variety of modalities and information entities, each one with its specifications and dependencies. The intrinsically complex nature of this scenario motivates the development of tools and methodologies capable of testing the compliance of produced DICOM objects with the standard. This article proposes and describes the development of a Web DICOM validation service for the medical imaging community that agglutinates, in a unique way, a set of functionalities. Its use can be as simple as uploading DICOM objects to be checked, without requiring platform registration or authentication, but ensuring data privacy by

* Corresponding author.
E-mail address: jorge.miguel.ferreira.silva@ua.pt (J.M. Silva).

https://doi.org/10.1016/j.csi.2018.06.003
Received 27 November 2017; Received in revised form 7 May 2018; Accepted 4 June 2018
0920-5489/ © 2018 Elsevier B.V. All rights reserved.

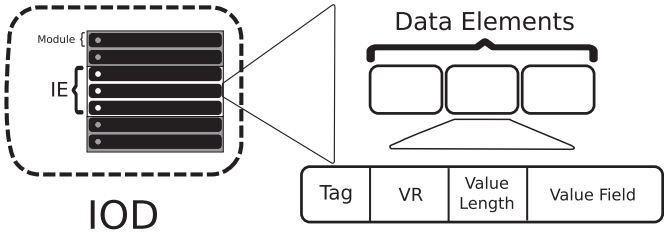

Fig. 1. Constitution of an IOD.

removing the patient's personal health information on the client side. Then, more complex validation tasks can be performed by the community in a collaborative way.

2. DICOM constitution

DICOM Information Model (DIM) rules the organization of information structures in the standard. It specifies the relationship between DICOM objects' information entities (IE) and real-world entities such as the patient, study, series and image. IE are computer data model abstractions for the real-word objects. Each IE may contain one or more Modules that are basic aggregations of related Data Elements (or Attributes). For instance, the patient module contains the name, ID, birth-date and sex attributes [9]. An object template is denominated as DICOM Information Object Definitions (IODs) and may contain one or more IE. IODs are normalized collections of DICOM Data Elements organized in Modules and IE (Fig. 1). The IOD Modules may be mandatory, conditional or user optional, as described in the DICOM Standard Part 3 [10]. Data Elements follow a TLV (Tag-Length-Value) encoding schema according to Part 5 of the standard [11]. The DICOM tag identifies an attribute using two hexadecimal numbers, called the group and element respectively. These numbers are specified as the ordered pair ($< group >$, $< element >$). For instance, Tag (0010,0010) identifies the Patient Name element of the Patient group. Length defines the size of the attribute's Value. The Value field contains the attribute's data. According to DICOM transfer syntax, an optional Value Representation (VR) element may also be present and specifies the attribute data type. There is also a Value Multiplicity that specifies the number of values that can be encoded in the value field of that Data Element [12]. The list of normalized Data Elements is defined in the DICOM Dictionary available in part 6 of the standard [13] and, according to the presence in Modules, are classified as:

- **Type 1**: Attribute presence is mandatory and must have a valid value;
- **Type 2**: Attribute presence is mandatory, but its value may be left blank;
- **Type 3**: Attribute presence is optional.

Furthermore, all types of attributes can be conditional (C), since IODs and Service-Object Pair (SOP) Classes, a combination of a DICOM service command (DIMSE) and an IOD, can define Data Elements that shall be included under certain specified conditions. Conditional types have the same requirements as their type (1, 2 or 3) under these conditions. As such, it is a protocol violation if the specified conditions are met, and the Data Element is not included. On the other hand, when the specified conditions are not met, Type C elements shall not be included in the dataset [11].

3. Related work

DICOM IODs are flexible structures and verification of objects' compliance with standard definitions may be a complex task. Due to the need to ensure the robustness and accuracy of software applications, programmers were the first to feel the need for verification tools. Then, healthcare IT staff requested end-user software applications to confirm the conformity of enterprise DICOM network nodes and debug abnormal events.

DCMCHECK [1] is commercial software that tries to solve this issue. It uses a specialized IOD description language which allows extensions (e.g. private elements, DICOM correction proposals) to be added to the IOD definition without changing the application itself [14]. The DICOM files are verified as conforming to the standard IOD definition (DICOM Part 3 [10])]), data structures and encoding (DICOM Part 5 [11]) and the data dictionary (DICOM Part 6 [13]). Furthermore, the DICOM File Meta Information (Preamble + DICOM Prefix + File Meta Information (0002, xxxx)) is evaluated according to the DICOM Part 10 specifications [15], as well as the consistency between it and the rest of the DICOM meta data information on the file.

DICOM Validation Toolkit (DVTk)[2] is an open source project for testing, validating and diagnosing problems with communication protocols in medical imaging environments [16]. It supports DICOM, HL7 and IHE integration profiles, and provides a DICOM Attribute Validator for validating DICOM files against definition files. The validator application includes GUI and command line versions, and a collection of .NET libraries for creating new validation and test tools. Moreover, it provides a DICOM Attribute Validator for validating DICOM files against definition files.

There are also other examples of open-source validation software like, for instance, the *dicom3tools/dciodvfy*[3] and the *dcm4che3* validator[4]. In general, they can check for inconsistencies in the DICOM files against Part 3 and 5 of the standard, Multiplicity against the Data Dictionary and Data Element Value content against encoding rules defined by the standard. Moreover, *dcm4che3* validator uses an undocumented XML file structure to determine the IOD structure and the mandatory Data Elements validation. The XML structure contemplates the IOD as the root element and the nested Data Elements. Furthermore, to enforce Value content validation against the VR attribute, Data Element can have an associated list of adaptable values that are useful for some attributes, for instance, the Patient's Sex (0010,0040). The XML file also supports conditional elements by using the clauses And, If and Or, which allow the definition of dependencies.

In terms of patents, an invention from 1997 [17] proposes an object-oriented structure that includes a plurality of semantic definition and validation objects, and a method that semantically validates the DICOM message by passing them through the structure and comparing the DICOM message to the provided definitions. In 2001, another patent [18] proposed a method for providing DICOM SR constraints within an XML document. To do so, the XML document was created containing DICOM SR constraints using declarative language. Moreover, a work from 2008 [19] proposes a technique that employs a XML validation document with a set of constraints specified for DICOM objects and makes use of them in validation processes.

Previously described validators are representative of the state-of-the-art in this field. They provide very useful functionalities but also have major limitations. First, these validators cannot resolve static preconditions that are dependent on the exam's protocol, rather than on the IOD itself. In other words, conditions that require input from the user to know how to validate the DICOM file. An example is the condition "C Required if contrast media was used in this image", which is present in many IODs. Secondly, the complexity of defining an entire configuration file for each IOD. This problem is aggravated by the first limitation since it creates the need to specify many configurations for the same IOD.

[1] DCMCHECK: http://dicom.offis.de/dcmcheck.php.en.
[2] DVTk: https://www.dvtk.org/.
[3] dicom3tools/dciodvfy: http://www.dclunie.com/dicom3tools/dciodvfy.html.
[4] dcm4che3: https://github.com/dcm4che/dcm4che.

Download English Version:

https://daneshyari.com/en/article/9951454

Download Persian Version:

https://daneshyari.com/article/9951454

Daneshyari.com