Informatics for Health: Connected Citizen-Led Wellness and Population Health
R. Randell et al. (Eds.)

doi:10.3233/978-1-61499-753-5-38

Web Validation Service for Ensuring Adherence to the DICOM Standard

Jorge Miguel SILVA [1], Tiago MARQUES GODINHO, David SILVA, Carlos COSTA

DETI/IEETA, University of Aveiro, Portugal

Abstract. The DICOM Standard has been fundamental for ensuring the interoperability of Picture Archive and Communications Systems (PACS). By compiling rigorously to the standard, medical imaging equipment and applications from different vendors can share their data, and create integrated workflows which contributes to better quality healthcare services. However, DICOM is a complex, flexible and very extensive standard. Thus, it is difficult to attest the conformity of data structures produced by DICOM applications resulting in unexpected behaviors, errors and malfunctions. Those situations may be critical for regular PACS operation, resulting in serious losses to the healthcare enterprise. Therefore, it is of paramount importance that application vendors and PACS administrators are confident that their applications follow the standard correctly. In this regard, we propose a method for validating the compliance of PACS application with the DICOM Standard. It can capture the intricate dependency structure of DI- COM modules and data elements using a relatively simple description language. The modular nature of our method allows describing each DICOM module, their attributes, and dependencies on a re-usable basis. As a result, our validator is able to encompass the numerous modules present in DICOM, as well as keep up with the emergence of new ones.

Keywords. DICOM, PACS, Medical Imaging, Validator

1. Introduction

Over the last decades, the use of digital medical imaging systems in healthcare institutions has increased remarkably [1]. Digital medical imaging systems are increasingly becoming central role tools for medical diagnosis and decision support [2]. Research and industry efforts to develop medical imaging equipment, including new acquisition modalities and information systems, are intense and have been grounded by the wide acceptance of the PACS concept [3]. It defines a set of hardware and software technologies that allow standardized data formats and communications between different equipment, applications and information systems [4]. PACS development has been supported by the DICOM Standard. It is the most universal and widespread standard used for the handling, storage, and transmission of digital medical images and related information [5].

The nowadays PACS ecosystem of applications and equipment resorts heavily to the exchange of medical imaging data between them [6]. Therefore, it is crucial that

[1] Corresponding author, Instituto de Engenharia Eletrónica e Informática de Aveiro, Campus Universitário de Santiago 3810-193 Aveiro, Portugal; Email: jorge.miguel.ferreira.silva@ua.pt

PACS components comply rigorously with the DICOM Standard. Since not doing so would compromise the entire operation of the PACS, resulting in potentially serious losses to the medical enterprise [7]. However, due to the complexity of DICOM and the wide variety of supported modalities and information entities, each one with its own set of specifications and dependencies, checking the compliance of an application is not trivial. Therefore, the necessity to conceive a method capable of verifying the compliance of DICOM Objects with the standard comes into sight. Our method checks if the attributes contained in DICOM files are per the standard, namely with the requirements defined in the DICOM Standard modules and templates denominated as Information Object Definition (IOD). It was deployed as a web-application where DICOM files can be uploaded and validated by the community.

2. Methods

DICOM is a Standard that specifies the information content, structure, encoding, and communications protocols for electronic interchange of diagnostic and therapeutic images and image-related information [8]. A central component in the DICOM Standard is the IODs [9], abstract data models used to specify real-world objects. They define which information must be included in each DICOM file, per its object type. Hence, there are IODs defined for different modalities such as the Magnetic Resonance image IOD or the Ultrasound Image IOD. The IOD definition includes a set of modules that contain information about a certain entity, for instance, Patients or Studies. The inclusion of these modules could be mandatory, conditional or optional. The information contained in each module is also defined in DICOM standard Part 3 [9].

Inside the modules, information is convoyed within data elements or attributes. These follow a TLV (Tag Length Value) structure and the standard provides a Data Dictionary that describes all possible attributes. There is also a VR (Value Representation) element that specifies the encoding of each attribute. These are 27 data types in DICOM. It defines the content type, including the characters allowed and prescribed data length. Besides the VR, each data element has also a multiplicity value which defines how many values the element may hold [10].

The wide variety and complexity of DICOM IODs, created the demand for the conception of software capable of automatically verifying attributes and organizations of DI COM files, per the requirements of IODs and Modules defined in the standard.

The dcm4che3 validator[2] is one of the best solutions that uses an undocumented XML file structure to assert IOD validation and verifies if mandatory attributes are defined or presented, as shown in this example 2. This structure consists of a root element; the IOD. The IODs children are the Data Elements, which have an associated Tag, VR just like the standard. This validator also supports the definition of a list of acceptable values for the data elements. This allows the definition of attributes such as the Patient's Sex (0010,0040) that must only contain one of' M"," F"," O". It is also capable of sup- porting conditional elements by using If, And, Or conditional clauses, allowing the definition of dependencies such as" Required if Responsible Person is present" in the Responsible Person Role attribute (0010,2298). The dicom3tools/dciodvfy is another solution for checking the DICOM objects

[2] https://github.com/dcm4che/dcm4che

conformance. Like dcm4che, it is capable of checking for inconsistent data inside attributes and between attributes[3].

Despite these features, these validators have two major limitations. First, it cannot resolve conditional elements such as" C - Required if contrast media was used in this image" that is present in many IODs, namely in the MR Image IOD. These are what we call static preconditions, which are not dependent on the IOD itself, but rather on the actual examination procedure. The second limitation is related with the complexity of defining an entire configuration file for each IOD. This problem is severely aggravated by the first problem because it may even be necessary to specify many configurations for the same IOD.

```
<IOD>
<Data Element keyword=" Attribute Name" tag =" xxxxyyyy" vr =" vv" type =" n"><! ––begin data element––>
<Value>Accepted Value 1</ Value>
<Value>Accepted Value 2</ Value>
</ Data Element> <! –– end data element ––>
</ IOD> <! –– end IOD definition ––>
```

3. Results

Our method tackles the problems mentioned above by enabling the definition of static preconditions that are resolved as inputs from the user. Moreover, it uses an enhanced configuration interface that defines both IODs Modules. By doing so, the validation software becomes able to support the re-usage of each module, and their combination to assemble the whole IOD definition. Lastly, it encompasses an online platform that promotes the sharing of modules and IOD definitions. This greatly reduces the necessity for defining new configurations and therefore simplifies the usage of the application.

As depicted in Figure 1, our methodology can be divided in 3 stages. The initial stage consists of a precondition processor. Preconditions, such as" Required if contrast media was used in this image", are defined in the module configuration file, and are handled by the application as a series of questions that are made to the user during the application's runtime. Furthermore, during this stage the different module configurations are combined into a composite IOD configuration automatically. The output of this stage is a customized IOD configuration file, like the dcm4che's IOD configuration file.

The next stage validates the DICOM files against the customized IOD definitions resultant from the first stage. In this process, we have leveraged the actual dcm4che3 validator, which we have extended to support new features. One may consider the customize IOD configuration resultant from the first stage as a super-set of the dcm4che3 IOD definitions. In this process, the data elements and their dependencies are checked in the source file.

The final stage performs a quality assurance validation. In this stage, some attributes in the DICOM file are checked if they comply with each other. For instance, if the Patient's Age matches the Study acquisition date, and the Patients Birth date. These validations are also defined in each modules configuration file.

[3] http://www.dclunie.com/dicom3tools/dciodvfy.html

Our validation system was deployed in a web platform to provide effortless access to a large user community as possible. This platform will also enable the community to create new, or improve existing module definitions.

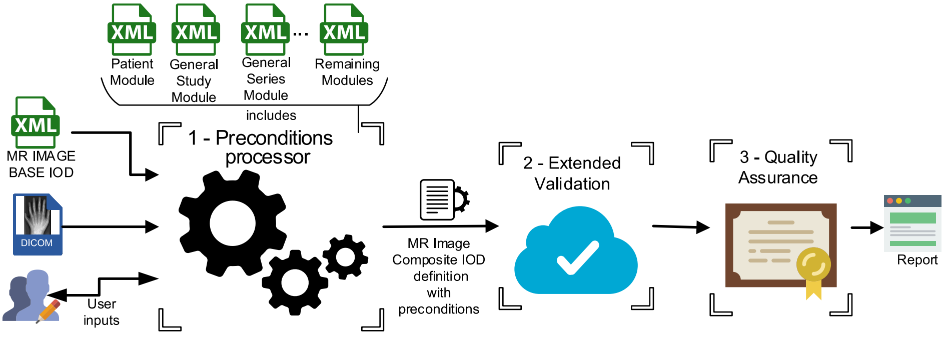

Figure 1. Methodology for the DICOM Validator.

4. Discussion and Conclusion

Our method provides a platform for validating DICOM files. This validator is intended for PACS administrators, developers as well as any other user with interest in validating their DICOM files. Its unique concept of community, and validation-as-a-service promises to greatly ease its usage. The possibility of testing preconditions, and reuse of definition in the DICOM modules level also increases its appeal and usability[4]. Figure 2 shows the result of a DICOM File validation of the Patient, study, series and Image modules. To compare it with other software we provided full access to our application where it is possible to test the validator with any DICOM file. The platform also includes some samples in the application portal, for those who might not have access to real DICOM images.

The implementation of an assurance validator will be useful for checking the congruity of the DICOM file between different attributes of the DICOM file. In other words, the validation method will consider the congruence of inter-attribute information. Finally, it is important to emphasize that this validator has novel features for validating data elements based on module configuration files. By making use of these files and the user's input, the validator creates customized IOD configuration file that defines which data elements and their dependencies are checked in the source file. This functionally increases the range of identifiable violations when compared to others re- ported in the literature.

As future work, we intend to develop more features to integrate in the stage 3, namely, check if different DICOM objects do or not belong to the same patient, and offer a way of fixing incorrect data.

[4] Demo at: http://bioinformatics.ua.pt/dicomvalidator

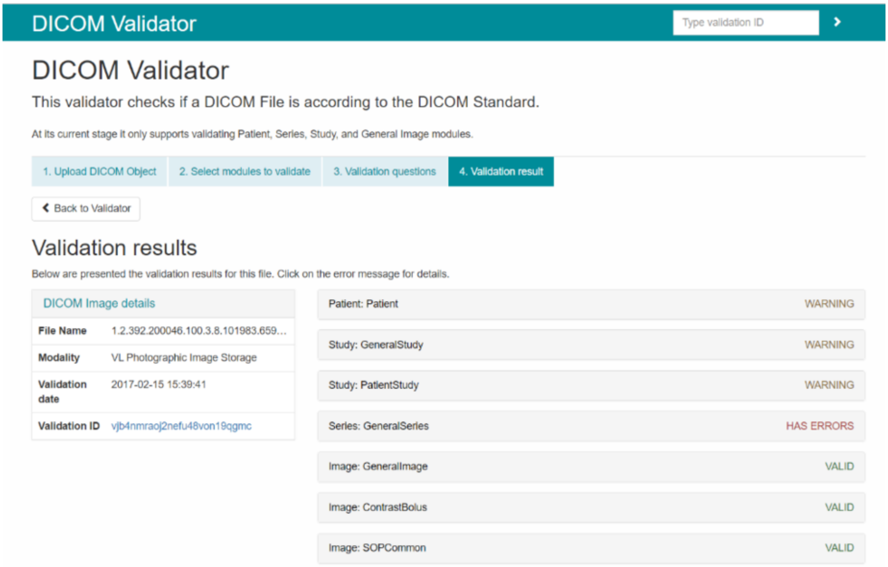

Figure 2. DICOM Validator.

Acknowledgments

Jorge Miguel Ferreira da Silva is funded by the Research grant Project CMUP- ERI / ICT / 0028/2014 - SCREEN-DR, Research Unit 127/94-IEETA. Tiago Marques Godinho is funded by Fundação para a Ciência e Tecnologia (FCT) under grant agreement SFRH/BD/104647/2014.

References

[1] Pedro Matos, Luis A Bastião Silva, Tiago Marques Godinho, and Carlos Costa. A Dynamic Approach to Support Interoperability for Medical Reports Using DICOM SR. Studies in health technology and informatics, 228:461–465, 2016.
[2] E Pinho, T Godinho, F Valente, and C Costa. A Multimodal Search Engine for Medical Imaging Studies. Journal of Digital Imaging, 2016:1–10, 2016.
[3] Frederico Valente, Luis A Bastião Silva, Tiago Marques Godinho, and Carlos Costa. Anatomy of an Extensible Open Source PACS. Journal of Digital Imaging, 29(3):284–296, 2016.
[4] H K Huang. PACS and Imaging Informatics: Basic Principles and Applications. Wiley, 2nd edition, 2010.
[5] Felix Fischer, M. Alper Selver, Sinem Gezer, Ouz Dicle, and Walter Hillen. Systematic parameterization, storage, and representation of volumetric DICOM data. Journal of Medical and Biological Engineering, 35(6):709–723, 2015.
[6] Andrew (Andrew G.) Webb. Introduction to biomedical imaging. Wiley-Interscience, 2003.
[7] Keith J. Dreyer, James H. Thrall, David S. Hirschorn, and Amit Mehta, editors. PACS. Springer-Verlag, New York, 2006.
[8] Oleg S. Pianykh. Digital Imaging and Communications in Medicine (DICOM) - A Practical Introduction and Survival Guide. Springer, 2012.
[9] NEMA. Digital Imaging and Communications in Medicine (DICOM) Part 3: Information Object Definitions.
[10] Herman. Oosterwijk and Paul T. Gihring. DICOM basics. OTech Inc, 2005.