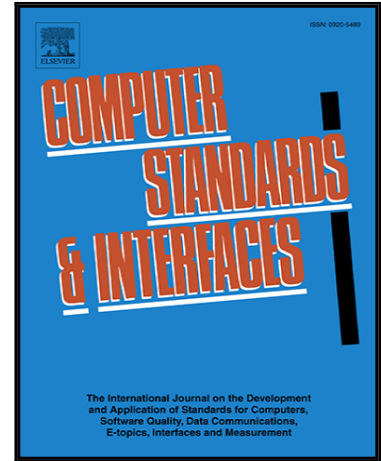


Accepted Manuscript

A Community-Driven Validation Service for Standard Medical Imaging Objects

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

PII: S0920-5489(17)30414-2
DOI: [10.1016/j.csi.2018.06.003](https://doi.org/10.1016/j.csi.2018.06.003)
Reference: CSI 3295



To appear in: *Computer Standards & Interfaces*

Received date: 27 November 2017
Revised date: 7 May 2018
Accepted date: 4 June 2018

Please cite this article as: Jorge Miguel Silva, Tiago Marques Godinho, David Silva, Carlos Costa, A Community-Driven Validation Service for Standard Medical Imaging Objects, *Computer Standards & Interfaces* (2018), doi: [10.1016/j.csi.2018.06.003](https://doi.org/10.1016/j.csi.2018.06.003)

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

Highlights

- Presents a Web DICOM validation service for the community
- Requires no configuration by the user
- Keeps up with new changes made to the DICOM standard
- Promotes validation results share-ability in the community
- Preserves patient data privacy since files are de-identified on the client side

ACCEPTED MANUSCRIPT

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

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.

Keywords: Web, DICOM, PACS, Medical Imaging, Validator

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, 2, 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 sys-

tems 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 removing the patients 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-world 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 (Figure 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 attributes Value. The Value field contains the attributes 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

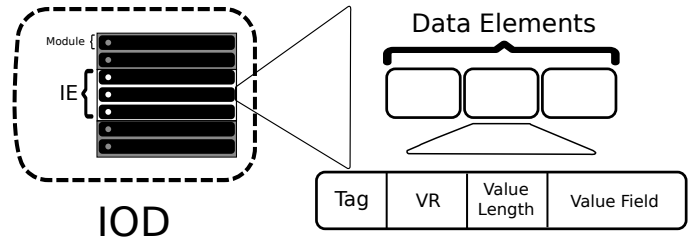


Figure 1: Constitution of an IOD

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* validator⁴. 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 Patients Sex (0010,0040). The XML file also supports conditional elements by using the clauses And, If and Or, which allow the definition of dependencies.

¹DCMCHECK: <http://dicom.offis.de/dcmcheck.php.en>

²DVTK: <https://www.dvtk.org/>

³dicom3tools/dciodvfy: <http://www.dclunie.com/dicom3tools/dciodvfy.html>

⁴dcm4che3: <https://github.com/dcm4che/dcm4che>

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 exams 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.

4. Proposed Architecture

4.1. Overview

The proposed platform provides a set of functionalities supported by state-of-the-art solutions but solves the precondition problem by creating an engine that uses the input from the user to deal with the preconditions and load the required modules. Since defining an entire configuration file for an IOD is extremely complex, a modular approach was followed where each module of the IOD file is defined in a nested separated file. This is done by using an enhanced configuration interface that defines both IODs and Modules. This architecture allows re-use of the modules when creating new IOD definitions. Aiming to support programmers and research teams from academy and industry, in a collaborative way, it was decided to incorporate the validation software into an online platform where users can create and share their modules and definitions. Firstly, this creates a dynamic and synergistic environment easily able to adapt to the creation of new modalities, and secondly, reduces the need to define new configurations and simplifies use of the application.

4.2. Functional Modules

The community-driven DICOM Validator has two main functionalities: Validation of DICOM files and support for the creation of description files, for Modality IODs and Modules, by the community. The description file is XML-based and specifies the Modules required for a given Modality IOD or what Attributes and respective Values are necessary for each Module.

The documentation regarding the structure of the definition file for both files is included in the platform, and can be accessed at <https://bioinformatics.ua.pt/dicomvalidator/#/docs>.

Figure 2 illustrates the validation use-case, where a users chosen file is uploaded to a remote server for validation. Before being sent, the file is locally modified (without overriding the users original one) to strip out pixel data and remove the patient's identifiable data for security and confidentiality reasons, as shown in Section 4.4.1

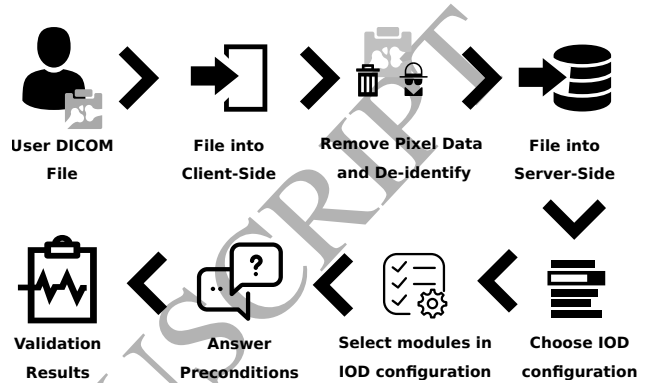


Figure 2: Platform validation process.

After receiving the file, the server loads the suitable description file for its Modality IOD and prompts the user to select which modules they want to validate. The user is then presented with several questions regarding the preconditions necessary for better validation results. A default answer to the questions can be provided. Finally, the validation results are shown. DICOM Validator provides a dashboard where users can follow up the validation processes. It also provides resources for the creation and editing of description files (Figure 3) using an embedded editor. These editing capabilities are required to make sure every description file is up-to-date and contains no errors. File editions are treated as contributions that can be commented on, and reviewed by the community before being accepted.

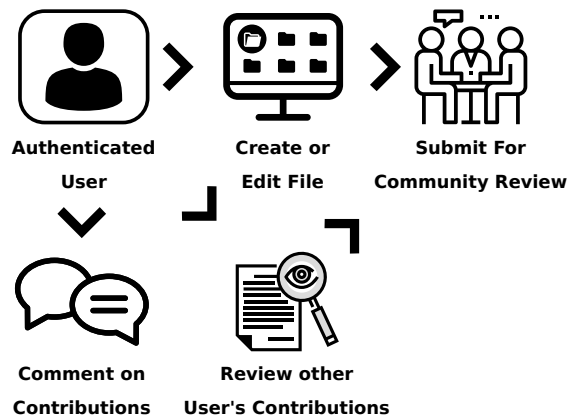


Figure 3: Process of creation of description files by the community.

The use of dashboard features is only accessible to authenticated members to ensure the community concept and tracing of actions. Anyone can register and sign-in to the DICOM

Validator platform. All authenticated users have access to the dashboard and can manually create or edit description files in the platform using a user-friendly editor, which are validated by the platform and can then be submitted as merge requests (pull requests). These merge requests are treated as contributions that can be commented on by the community before being accepted. Platform's administrator has the responsibility of accepting and merging these changes with the master repository to ensure the quality standards of the definition files. The authentication system is handled by third-parties such as Google, GitHub, or Dropbox by using the *OAuth* protocol⁵.

4.3. Software layers

DICOM Validator consists of three software layers: a front-end, a back-end and a dashboard which encompasses a collaborative platform for editing description files.

The front-end is built using *Angular*⁶ and *Bootstrap*⁷, communicating with the back-end module through a REST API. It also includes a de-identification tool developed with *dcmjs*⁸. The back-end module was built using *Jetty*⁹ to ensure communication, and additional Java Classes to perform validation, module loading, and user session logic. This module also stores the latest versions of the description files required by the validator. Lastly, a modified version of *Gitea*¹⁰ was used to construct the dashboard. The changes include support for communication with the back-end and XML Editing using *Xonomy*¹¹.

As Gitea is backed by a Git Server, that allows us to take advantage of this feature to keep the back-end description files up-to-date by syncing their respective folders with their repositories master branch as shown in Figure 4.

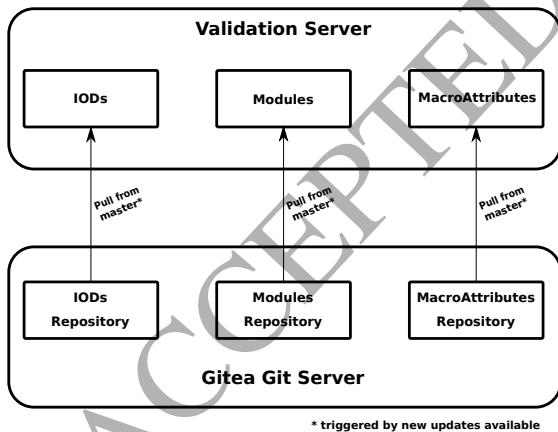


Figure 4: Platform Creation or Editing of Modality IOD or Module Configurations Use-Case.

⁵OAuth 2.0: <https://oauth.net/2/>

⁶<https://angular.io/>

⁷<http://getbootstrap.com/>

⁸dcmjs: <http://dcmjs.org/>

⁹<http://www.eclipse.org/jetty/>

¹⁰<https://gitea.io>

¹¹<http://www.lexiconista.com/xonomy/>

4.4. Workflows

4.4.1. Pixel Data Removal and De-Identification

Typically, the DICOM file element requiring most storage space is the pixel data. For instance, the extreme case of DICOM objects for pathology imaging may require dozens of gigabytes [20, 21]. Since the validation process does not request analysis of visual information, removal of pixel data on the client-side has the advantage of reducing the files upload time as well as the storage space required at the server. Furthermore, by removing pixel data, we are also eliminating any PHI that can be burned into images of some modalities such as in Ultrasound (US) and External-camera Photography [22].

DICOM standard PS3.15 Annex E provides the standard de-identification guidelines [23]. Table 3 in the Additional Material provides comprehensive information regarding actions that the DICOM standard recommends for the Basic Profile attributes that we de-identified, and examples of what we provided as value for that attribute. However, our de-identification mechanism is not completely compliant with the standard, since we intended to create a fast mechanism that could easily replace the values of the attributes with dummy values consistent with the original attribute's value in terms of length and data type. This was done to ensure that we would not affect the DICOM object's validation regarding its structure. The unique objective is to use the data in the validation process, not the archiving or sharing of DICOM anonymized data.

The proposed platform uses *dcmjs*, a Javascript cross-compilation of *dcmk*¹², to transform the input DICOM file into a representative XML file. The pixel data is removed in this transcoding process and the metadata is de-identified to ensure patient privacy and confidentiality [24]. The de-identification process is done by reading the XML elements and replacing the fields containing the patients personal health information (PHI). The replacement (anonymous) sequence has the same length and domain type of original data, and is in accordance with the lists of adaptable values for that Data Element (see Section 5). For example, a *PatientName* attribute with the content *DoeJohn* would become *REMOVEDR* and a *PatientAge* with 26 would become *00*. Once this process is concluded, the XML file is transformed again into DICOM and passed to the server for validation.

5. Validation Process

Our web interface lies in a Model-View-Controller (MVC) paradigm and the framework works following the Service-Oriented Architecture (SOA), where the client-server communication is performed via Restless services. Our services are stateful and can be consumed by third party. The services exposed by our API are the following:

- **/configure**: endpoint where files are uploaded to the platform and, as response, it is returned a validation ID and a set of modules and options.

¹²dcmk: <http://dicom.offis.de/dcmk.php.en>

- **/validate:** to provide the answer to /configure service. For instance, which modules will be validated, as well as the usage (or not) of the default answers for the preconditions. It returns the precondition questions or the files validation result if the user opted by the default answers to the preconditions.
- **/result:** receives a validation ID and a response to the precondition questions (optional); and returns the files validation results.

The validation process is released on the server side. When a file reaches the applications server, the DICOM metadata is read to obtain the *SOPClassUID*, which is used to identify the medical imaging modality and select the respective description file to be used in the validation process. The selection of an appropriate description file is essential to obtain reliable results, since distinct DICOM modalities have different requirements. An IOD description file is made by including the modules and preconditions that the user must answer. The preconditions can be placed either at the level of the IOD or the module depending on where they will be used (i.e., for inclusion of Modules or Data Elements, respectively). Listing 1 shows part of the IOD description file for the CR Modality IOD. Included Modules are defined in the file by their Information Entity (IE), module name and use (Mandatory (M), Conditional (C) or User Defined (U)). A precondition is defined with an id name, a default value and a question that will be asked to the user during the validation process.

Listing 1: Platform IOD configuration example for CR modality

```
<IOD>
<define id="contrasMediaWasUsed" value="true"
  question="Was contrast media used in this Image">
<include ie="Patient" module="Patient" usage="M" />
<include ie="Study" module="GeneralStudy" usage="M" />
<include ie="Study" module="PatientStudy" usage="U" />
<include ie="Series" module="GeneralSeries" usage="M" />
<include ie="Image" module="GeneralImage" usage="M" />
<include ie="Image" module="ContrastBolus" usage="M" >
<If idref="contrasMediaWasUsed" />
</include>
<include ie="Image" module="SOPCommon" usage="M" />
</IOD>
```

Besides preconditions, Module description files may include Macro Attribute tables and contain the Data Elements in accordance with the standard (Listing 2). Each Data Element is categorized by a keyword, a tag, a VR, a VM, a type and the number of items (only if VR is a Sequence (SQ)). Furthermore, each Data Element can have an associated list of acceptable values.

Listing 2: Platform Module example of a Definitions XML file for the Patient Module

```
<Module>
<define id="PatientIsAnAnimal"
```

```
  question="Is patient an animal?"
  default="false" />
<define id="ResponsiblePersonIsPresent"
  question="Is the responsible person present?"
  default="false" />
<DataElement
  keyword="PatientName"
  tag="00100010"
  vr="PN" type="2" vm="1" />
<DataElement
  keyword="PatientID"
  tag="00100020"
  vr="LO" type="2" vm="1" />
<include table="10-18" />
<DataElement
  keyword="PatientBirthDate"
  tag="00100030"
  vr="DA" type="2" vm="1" />
<DataElement
  keyword="PatientSex"
  tag="00100040"
  vr="CS" type="2" vm="1" >
  <Value>M</Value>
  <Value>F</Value>
  <Value>O</Value>
</DataElement>
...
</Module>
```

Platform users may be questioned for two reasons: to select the modules to be validated and to answer preconditions. Depending on the users actions, modules may be excluded or interpreted in diverse ways. Description file parsing is made using a custom XML handler based on *dcm4che3s SAXHandler*¹³ and data validation is made using *dcm4che3s* validator. Since *dcm4che3* validator natively ignores the validation of the Data Elements of type 3, it was necessary to adapt its behaviour to add support for validation of elements of this type. Moreover, the proposed validator extends the *dcm4che3* validator by adding additional conditions, preconditions and a more modular approach to inclusion of the module in the IOD description files. The results of the validation process are shown in 5 types of categories regarding the validation of each module:

- **Valid**, when all of the modules content is in accordance with the standard;
- **Warning**, when Data Elements of type 3 are not compliant with the standard;
- **Skipped**, when the user chooses not to select a specific IOD module for validation;
- **Unsatisfied Condition**, when a requested condition for validating a module was not met;
- **Has Errors**, when Data Elements of type 1 and 2 are not in accordance with the standard.

All the messages exchanged between the client and server modules (e.g. user's answer to questions and results of validation process) are based on JSON object.

¹³<https://github.com/dcm4che/dcm4che/blob/master/dcm4che-core/src/main/java/org/dcm4che3/data/IOD.java>

6. Results

6.1. Management Dashboard

The Validators Dashboard page allows users to browse and edit the contents of Modules, Modality IODs and Macro Attribute Table repositories (Figure 5).

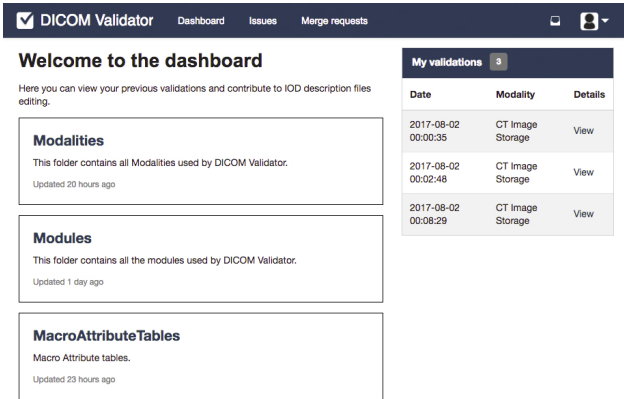


Figure 5: DICOM Validator Dashboard.

The user can view the existing definition files for the Modality IODs (Figure 6) and Modules. The dashboard can show the user the history of their previous validations. Since the solution is back-end by a Git server, the platform can track the updating of description files by the different community members, accepting (merging) or rejecting the contribution requests, controlling in this way the quality of definition files used in production. Before submitting a new (file) contribution, the user must specify what changes were made by providing a title for the contribution and a description. The community can comment on the changes and suggest improvements, and the platform administrator can even approve the changes, merging the new code with the existing one in the master workspace and triggering the back-end validation service to update its description files to support the new ones.

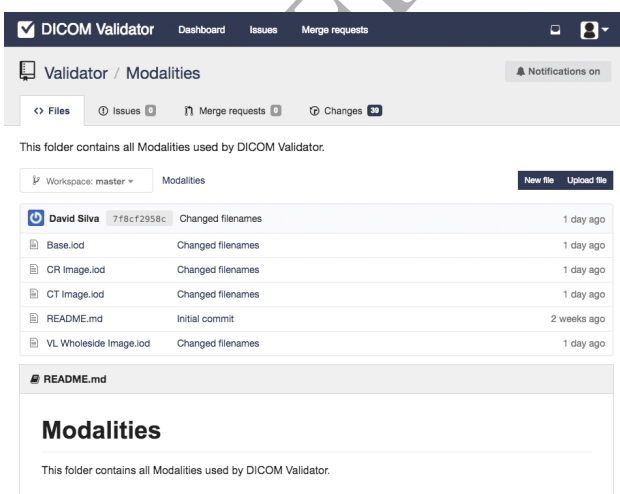


Figure 6: View of existing definition files for the Modality IODs.

A description file can be edited in the platform using a user-friendly editor (Figure 7). This facilitates the creation of description files by automating several important tasks. Besides editing, the user can open issues for situations that may be happening, and view changes made by other users.

After the editing is finished, the changed file is submitted to the back-end service, where the validator evaluates this new contribution and enumerates eventual errors found while parsing the description file. To do so, it first tries to identify the content as belonging to a file definition of a module or IOD. Once identified, it attempts to convert the XML to an IOD Module object or an IOD object, respectively. If the build fails, it is because the file structure is wrong, and an error is reported to the user. The user can then identify in which lines an error is present and correct them. If the validator ensures there are no errors, the description file can be submitted for community review.

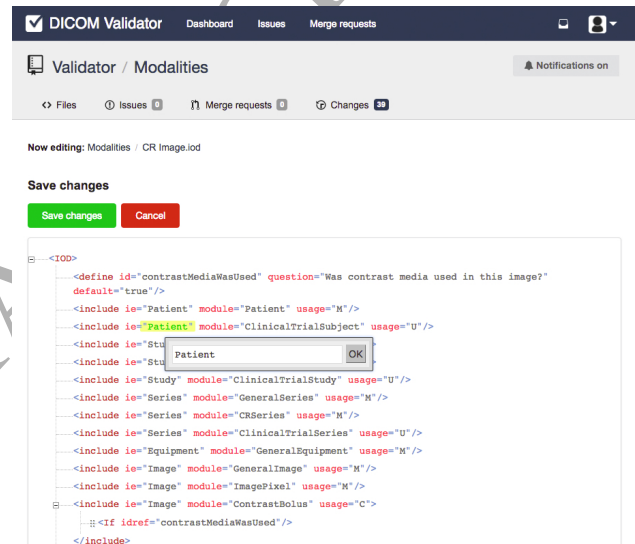


Figure 7: Edition of the definition file for the CR Modality IOD.

6.2. Validation Service

Our proposed platform can be easily accessed via the URL <https://bioinformatics.ua.pt/dicomvalidator/>.

To evaluate the validation service, 3 DICOM files from distinct modalities were used and originated from a publicly available dataset:

- A Computed Radiography (CR) modality, from Belarus Tuberculosis Portal (*Belarus TB*)¹⁴
- A Computed Tomography (CT) from the *DICOM Library*¹⁵
- A Whole Slide Imaging (WSI) from *DICOM WG26 - Pathology*¹⁶

¹⁴Belarus TB: <http://tuberculosis.by/>

¹⁵DICOM Library: <http://www.dicomlibrary.com/>

¹⁶Nema: <ftp://medical.nema.org/medical/dicom/DataSets/WG26/Hamamatsu/>

For each of the previous samples, the corresponding IOD definition and module files were created and made available in the DICOM Validator platform. Next, the files went through the pipeline described in Figure 2, where preconditions were answered as default. The validation results are presented in the Figure 8 and can be consulted by clicking on the validation codes shown in Table 1.

Table 1: Validation ID of for each modality file validated.

Modality	Validation Code
CR	8r5vl6hcsmljoendga4jfiqud
CT	593cik8vd9p4n3t0g7f172j1lq
WSI	8cvjgffoat1ivin10stk6u7i8r

Each validated module is classified using one of the categories presented in Section 5 (Valid, Has Errors, Warning, Skipped and Unsatisfied Condition). The results can be displayed as a summary or in a detailed view. The latter shows the attributes that are missing or have invalid values. While analyzing other validators we noticed a lack of warnings regarding Type 3 attributes. Although they are legal, our line of reasoning was that their presence or absence as well as if they are in accordance with the standard should be displayed to the user. However, the display of such results may make it hard to distinguish important from unimportant results. As such, the platform provides a checkbox list that allows the users to select what types of error they wish to view after the validation. An important aspect is that, after the validation, the user can go directly to *DICOMLookup*¹⁷. This provides the ability to link to context-sensitive information about each error. However, the *DICOMLookup* service is not normative, nor necessarily maintained nor current. Therefore, we added a link directly to the CHTML online DICOM standard itself.

7. Discussion

7.1. Relevance of the Work

The acceptance of DICOM standard by medical industry and researchers has brought to the market many devices that are sold as being DICOM conformant. However, in practice, the interoperability between different DICOM equipment is not always accomplished easily and completely. For instance, a study evaluated the DICOM conformance of cardiac X-ray Angiography objects from ten vendors using two validation tools and five visualization applications [25]. The results showed that only two datasets were completely conformant with the DICOM standard and three contained serious errors that inhibited their correct visualization. Another important example was performed at the German Congress of Radiology 2006, where radiologists were invited to bring their CD for a short test against the CD specification published by German Society of Radiology, and 80% of those real world CDs failed the test [26].

This file has inconsistencies!

Validation Code: 8r5vl6hcsmljoendga4jfiqud
Modality: Computed Radiography Image Storage
Validation Date: 2017-08-02 18:36:49

[Go back](#)

Module details

IE: Patient Module: Patient WARNING +

Type 1 Attributes: No issues.
Type 1C Attributes: No issues.
Type 2 Attributes: No issues.
Type 2C Attributes: No issues.
Type 3 Attributes: 9 attribute(s) missing;

IE: Patient Module: ClinicalTrialSubject HAS ERRORS +

Type 1 Attributes: 2 attribute(s) missing;
Type 1C Attributes: 2 attribute(s) missing;
Type 2 Attributes: 3 attribute(s) missing;
Type 2C Attributes: No issues.
Type 3 Attributes: 1 attribute(s) missing;

This file has inconsistencies!

Validation Code: 593cik8vd9p4n3t0g7f172j1lq
Modality: CT Image Storage
Validation Date: 2017-08-02 18:36:39

[Go back](#)

Module details

IE: Patient Module: Patient HAS ERRORS +

Type 1 Attributes: No issues.
Type 1C Attributes: No issues.
Type 2 Attributes: 2 attribute(s) missing;
Type 2C Attributes: No issues.
Type 3 Attributes: 9 attribute(s) missing;

IE: Patient Module: ClinicalTrialSubject HAS ERRORS +

Type 1 Attributes: 2 attribute(s) missing;
Type 1C Attributes: 2 attribute(s) missing;
Type 2 Attributes: 3 attribute(s) missing;
Type 2C Attributes: No issues.
Type 3 Attributes: 1 attribute(s) missing;

This file has inconsistencies!

Validation Code: 8cvjgffoat1ivin10stk6u7i8r
Modality: VL Whole Slide Microscopy Image Storage
Validation Date: 2017-08-02 18:37:40

[Go back](#)

Module details

IE: Patient Module: Patient WARNING +

Type 1 Attributes: No issues.
Type 1C Attributes: No issues.
Type 2 Attributes: No issues.
Type 2C Attributes: No issues.
Type 3 Attributes: 9 attribute(s) missing;

IE: Patient Module: ClinicalTrialSubject HAS ERRORS +

Type 1 Attributes: 2 attribute(s) missing;
Type 1C Attributes: 2 attribute(s) missing;
Type 2 Attributes: 3 attribute(s) missing;
Type 2C Attributes: No issues.
Type 3 Attributes: 1 attribute(s) missing;

Figure 8: Sample of the validation results obtained for the CR, CT and WSI files, showing for each file the errors occurred in each module ordered by the Data Element Type.

Table 2: Qualitative comparison analysis between related works.

Validation Software	Open Source	Web Based	De-identification	Preconditions	Consistency between DICOM objects	Update Capability
dicom3tools/dciodvfy	✓	×	×	×	×	×
dicom3tools/dcentvfy	✓	×	×	×	✓	×
DVTK	✓	×	×	×	×	×
dcm4che3	✓	×	×	×	×	×
DCMCHECK	×	×	×	×	×	×
DICOM Validator	✓	✓	✓	✓	×	✓

7.2. Comparison with State-of-the-Art and Solution Limitations

Table 2 compares the different State-of-the-art solutions with proposed DICOM Validator.

Currently, neither the DICOM Standards Committee nor MITA (NEMA) have any official tool or certification mechanism. However, evidence points to the fact that the existence of validation tools is fundamental to ensure interoperability between systems, since the dciodvfy tool and DVTK are used in IHE Connectathons for helping in the evaluation of systems interoperability [27].

As it can be observed in Table 2, most state-of-the-art solutions possess limitations that are covered by our solution, namely the fact that these validators are not web based, cannot resolve static preconditions and have a limited capability of keeping updated with the standard. On the one hand, the fact that our system can run on a Web browser allows the user to perform simple tests without the need to deploy the system locally. On the other hand, the ability to resolve preconditions, as well as to host a community, provides users with more reliable validation results whilst maintaining the platform up to date with the standard. The dciodvfy tool was Internet-deployed, but it is no longer available for the community. It used a CGI script on a web server that returned the output of an executable as if it runs in a command line. The tool did not support the de-identification of the images prior to uploading nor provided any mechanism for specifying predicates or to address real world conditional requirements. Another important aspect is that the majority of these validators do not perform de-identification of PHI tags. Relatively to the platform trust model, there are no benefits from sending the identifiable object to the server. Independently of the trust in the organization that operates the server, it is always preferred to not share sensitive data with third entities, inclusively by legal restrictions [28]. The user feels more comfortable to use a service with this *modus operandi*. Moreover, the de-identification at client-side is an automatic process, not being a burden to the user. Finally, by removing the pixel data, the volume of data and consequent upload time is significantly reduced, which is fundamental to improve user experience.

Furthermore, there are some validators such as DVTK and dicom3tools/dcentvfy that support the evaluation of consistency between DICOM objects. Although this feature is important,

it is worth noting that our validator was designed to validate individual DICOM objects and not to evaluate or compare the consistency between DICOM objects. As such, to perform such tasks one requires the use of the priorly mentioned validators.

7.3. Future Work

Although we covered a large amount of requirements in order to create a solid, updated system, there are some aspects that we hope to tackle in future developments of our work.

Firstly, although the online service provided by the validator platform can be useful for some usage scenarios, developers may also request bulk verification that runs locally as well as its incorporation in automated regression testing frameworks. As such, one good improvement to this work will be the development of a command line extension that will consume the definition files created by the Validator's online community.

Secondly, a side-effect associated to the de-identification of identifiers attributes is that we are not able to validate them at server side. A possible solution is the user select to bypass the de-identification mechanism. Removing pixel data is useful in terms of efficiency, since we are reducing the files upload time as well as the storage space required at the server. However, by doing so, proposed solution is failing to validate the actual pixel data, regarding its length and encoding. Currently, the platform maintains the number of frames of the file but does not consider the encoding and consequentially the length. As such, one possible improvement would be to upload pixel data at the expense of performance or to pass this information to the validator by extracting the information from the pixel data (7FE0,0010) prior to remove it at client side of the application, and then compare these values to the ones present in the meta-data of the file.

Another important remark regarding the validators design was the use of preconditions. Although useful, answering all the precondition questions can be a quite tiresome and time-consuming task. On the other hand, leaving the selection to default can lead to the creation of many errors. As such, in future work, one could improve this work by creating a Management Dashboard feature that could let the user specify the granularity of the preconditions the user intends to respond.

Finally, another future improvement would be to provide users with a tool that could parse the DocBook XML of DICOM standard automatically and build our XML files. This tool would decrease the amount of user effort when creating new description files.

¹⁷DICOMLookup: <http://dicomlookup.com/>

8. Conclusion

Digital medical imaging laboratories rely greatly on DICOM standard to ensure interoperability between different equipment. However, manufacturers' implementation of this standard may have non-conformities at several levels and medical staff may be responsible for data inconsistencies when entering data. The capacity to validate the quality of data and its compliance with the DICOM standard is a fundamental issue to avoid disruption in services. This article presents an innovative community-driven web validation service for DICOM files. It runs in a common Web browser and can be safely used to validate real-world files since they are de-identified on the client side. The community can contribute to improving the platform by creating or editing description files used in the validation process. This means the platform can be updated whenever a new change is made to the DICOM standard or to resolve any issue that may produce unsatisfactory results and lead to unintended mistakes by the user. These contributions are always guided by a graphic interface tool and final verification is ensured before being submitted for the community's approval, meaning description files are less prone to errors. The results of file validation are saved on the database and can be accessed and shared by the user. In conclusion, we propose a solution that solves problems faced by other state-of-the-art validators and is also prepared to evolve along with the DICOM standard.

Acknowledgments

Jorge Miguel Silva is funded by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation COMPETE 2020 Programme, and by National Funds through the FCT Portugal (Fundação para a Ciência e a Tecnologia) within project CMUP-ERI/ICT/0028/2014 SCREEN-DR. Tiago Marques Godinho is funded by FCT under grant agreement SFRH/BD/104647/2014.

References

- [1] H. K. Huang, *PACS and Imaging Informatics: Basic Principles and Applications*, Wiley, 2nd edition, 2010.
- [2] C. Creighton, A literature review on communication between picture archiving and communication systems and radiology information systems and/or hospital information systems, *Journal of Digital Imaging* 12 (1999) 138–143.
- [3] T. J. St. Cyr, An overview of healthcare standards, in: 2013 Proceedings of IEEE Southeastcon, IEEE, 2013, pp. 1–5.
- [4] P. Mildenerger, M. Eichelberg, E. Martin, Introduction to the DICOM standard, *European Radiology* 12 (2002) 920–927.
- [5] A. A. G. Webb, *Introduction to biomedical imaging*, Wiley-Interscience, 2003.
- [6] Realization of integration and working procedure on digital hospital information system, *Computer Standards & Interfaces* 25 (2003) 529–537.
- [7] K. J. Dreyer, D. S. Hirschorn, J. H. Thrall, A. Mehta, *PACS: A Guide to the Digital Revolution*, Springer New York, 2006.
- [8] NEMA, *Digital Imaging and Communications in Medicine (DICOM) Part 2 : Conformance*, 2018.
- [9] S. C. Horii, DICOM, in: S. G. L. George C. Kagadis (Ed.), *Informatics in Medical Imaging*, CRC Press, 2011, pp. 48–74.

- [10] NEMA, *Digital Imaging and Communications in Medicine (DICOM) Part 3 : Information Object Definitions*, 2018.
- [11] NEMA, *Digital Imaging and Communications in Medicine (DICOM) Part 5: Data Structures and Encoding*, 2018.
- [12] H. Oosterwijk, P. T. Gihring, *DICOM basics*, OTech Inc, 2005.
- [13] NEMA, *Digital Imaging and Communications in Medicine (DICOM) Part 6 : Data Dictionary*, 2018.
- [14] A. J. Hewett, H. Grevenmeyer, A. Barth, M. Eichelberg, P. F. Jensch, Conformance testing of DICOM image objects, in: S. C. Horii, G. J. Blaine (Eds.), *Proceedings of the SPIE, Volume 3035*, p. 480–487 (1997)., volume 3035, pp. 480–487.
- [15] NEMA, *Digital Imaging and Communications in Medicine (DICOM) Part 10: Media Storage and File Format for Media Interchange*, 2018.
- [16] G. Potter, R. Busbridge, M. Toland, P. Nagy, Mastering DICOM with DVTK, *Journal of Digital Imaging* 20 (2007) 47–62.
- [17] T. M. B. C. C. M. D. G. T. Tian Helen He (Plano, Method for validating a digital imaging communication standard message, 1997.
- [18] N. Y. U. Lee Kwok Pun (Flushing, Specifying DICOM semantic constraints in XML, 2001.
- [19] N. H. U. Guo Dongbai (Nashua, Techniques for checking whether a complex digital object conforms to a standard, 2008.
- [20] R. Singh, L. Chubb, L. Pantanowitz, A. Parwani, Standardization in digital pathology: Supplement 145 of the DICOM standards., *Journal of pathology informatics* 2 (2011) 23.
- [21] An efficient architecture to support digital pathology in standard medical imaging repositories, *Journal of Biomedical Informatics* 71 (2017) 190–197.
- [22] J. M. Silva, E. Pinho, E. Monteiro, J. F. Silva, C. Costa, Controlled searching in reversibly de-identified medical imaging archives, *Journal of Biomedical Informatics* 77 (2018) 81–90.
- [23] NEMA, *DICOM standard PS3.15 Annex E*, 2018.
- [24] T. Douraki, Ethical and Legal Dimensions of Medical Confidentiality in European Law of Human Rights, in: *Manage or Perish?*, Springer US, Boston, MA, 1999, pp. 421–427.
- [25] Maarten H. Baljon, Maarten G. Gerritsen, Marco Eichelberg, Peter Jensch, Quality Control using Automated Validation Tools can Improve Interoperability of DICOM Implementations, 1999.
- [26] M. Onkena, M. Eichelberga, J. Riesmeierb, P. Mildenergerc, Image distribution and integration strategies - Exchange of radiological images on DICOM CD: a survey of the state of technology in Germany, *International Journal of Computer Assisted Radiology and Surgery* 2 (2007) 290–292.
- [27] D. Clunie, *DICOM Validator - dciodvfy*, 2015.
- [28] M. Sariyar, I. Schluender, C. Smee, S. Suhr, Sharing and Reuse of Sensitive Data and Samples: Supporting Researchers in Identifying Ethical and Legal Requirements., *Biopreservation and biobanking* 13 (2015) 263–270.