XML-Based DICOM Data Format

Cong Yu¹ and Zhihong Yao^{2,3}

To enhance the readability, improve the structure, and facilitate the sharing of digital imaging and communications in medicine (DICOM) files, this research proposed one kind of XML-based DICOM data format. Because XML Schema offers great flexibility for expressing constraints on the content model of elements, we used it to describe the new format, thus making it consistent with the one originally defined by DICOM. Meanwhile, such schemas can be used in the creation and validation of the XML-encoded DICOM files, acting as a standard for data transmission and sharing on the Web. Upon defining the new data format, we started with representing a single data element and further improved the whole data structure with the method of modularization. In contrast to the original format, the new one possesses better structure without loss of related information. In addition, we demonstrated the application of XSLT and XQuery. All of the advantages mentioned above resulted from this new data format.

KEY WORDS: DICOM, data format, XML, XML Schema

INTRODUCTION

Digital imaging and communications in medicine (DICOM)¹ was raised with the development of medical imaging informatics. It creates a public and license-free standard method for the transmission of digital medical images and their associated information. Currently, DICOM has been widely adopted by hospitals since it enables the integration of scanners, servers, workstations, printers, and network hardware from multiple manufacturers into a picture archiving and communication system (PACS). In DICOM standard, there are many separate parts and each part of the standard concentrates on a different point of view of the DICOM protocol. The main interest in this research was focused on part 3 (Information Object Definitions) and part 5 (Data Structures and Encoding).

The Data Structures and Encoding, part 5 of DICOM standard specifies how DICOM applications construct and encode the Data Set. A Data Set (Fig. 1) is constructed of Data Elements, which consists of four parts: Data Element Tag, Value Representation (VR), Value Length and Value Field.

- Data Element Tag: an ordered pair of 16-bit unsigned integers representing the group number followed by the element number, like (XXXX,YYYY). It uniquely identifies the Data Element.
- VR: A two-byte character string, which describes the data type and format in Value Field
- Value Length: a 16 or 32-bit unsigned integer, which shows the length of Value Field
- Value Field: the stored information

DICOM data model is described by information object definition (IOD). As is shown in Table 1, an

¹From the Biomedical Engineering, School of Medicine, Shanghai Jiao Tong University, No.227 South Chongqing Road, Shanghai, People's Republic of China.

²From the Institute of Health Sciences, Shanghai Institute of Biological Sciences, Chinese Academy of Sciences, Room 504, Building 1, No.225 South Chongqing Road, Shanghai, People's Republic of China.

³From the Computer Science, Ruijin Hospital affiliated to Shanghai Jiao Tong University, No.197 Rui Jin Er Road, Shanghai, People's Republic of China.

Correspondence to: Zhihong Yao, Institute of Health Sciences, Shanghai Institute of Biological Sciences, Chinese Academy of Sciences, Room 504, Building 1, No.225 South Chongqing Road, Shanghai, People's Republic of China; tel: +86-21-63049429; e-mail: rj_yzh@hotmail.com

Copyright © 2009 by Society for Imaging Informatics in Medicine

Online publication 28 January 2009

doi: 10.1007/s10278-008-9173-5

Fig. 1. DICOM data structure.

IOD is a collection of related pieces of information, grouped in information entities (IE). For each IE, the second column lists some Modules that belong to it. Modules are defined in another section of the standard, and they can be used in more than one IOD. The last column indicates whether the use of Module is mandatory, conditional or user option.

In Modules (Table 2), related information are grouped, and they are called Attribute. An Attribute is encoded as a Data Element, may or may not be required in a Data Set, depending on its Type. There are five Types, including 1, 1C, 2, 2C, and 3.

From the DICOM data format, the Data Element Tag is simple and unambiguous, however, its readability is limited and only standard Data Element can be understood (through looking up Data Dictionary). For this reason, we proposed to introduce XML into DICOM,² replacing numeric tag with XML character string tag, thus making it possess explicit meaning. Another limitation of DICOM is that the Data Element is stored in linear structure in a Data Set, which becomes a great barrier for expressing and processing the rich structured content of image data and may result in poor searching efficiency. However, this barrier can be overcome by mapping DICOM to XML,³ for XML is especially well suited to expressing the

Table 2. Patient Module

Attribute name	Tag	Type	Attribute description
Patient's name	(0010,0010)	2	–
Patient ID	(0010,0020)	2	–
Issuer of patient ID	(0010,0021)	3	–
Patient's birth date	(0010,0030)	2	–
Patient's sex	(0010,0040)	2	–
–			

Type 1 required, Type 1C conditional, Type 2 required, Type 2C conditional, Type 3 optional

structured content. Meanwhile, since an open format can guarantee the access to data without legal or technical barriers, mapping DICOM to XML could make medical images accessible in a long time. In addition, to ensure that the data structure could actually conform to a unified standard, we designed XML Schema to validate the XML-based data format, thus some errors such as inconsistent naming or incorrect structure can be avoided.

METHODOLOGY

The W3C XML Schema Definition Language is an XML language for describing and constraining the content of XML documents. It includes extensive support for type inheritance and enables the reuse of previously defined structures, which offers great flexibility for describing the complex DICOM data format.

DICOM Data Type

In DICOM standard, the type and format of Data Element are decided by VR, and the definition of VR is in text-form. For example, “AS” (Age String) is defined as a string of characters with one of the following formats: nnnD, nnnW, nnnM, nnnY, where nnn shall contain the number of days for D, weeks for W, months for M, or years for Y. Although the definition is in detail, it is not so concise and sometimes maybe mistakenly understood. However, it is easy to describe it with XML Schema. To express “AS”, we declared a derived data type to

Table 1. CT Image IOD Module

IE	Module	Reference	Usage
Patient	Patient	C.7.1.1	M
	Clinical trial subject	C.7.1.3	U
	Patient study	C.7.2.2	U
Study	General study	C.7.2.1	M
	Clinical trial study	C.7.2.3	U
	General series	C.7.3.1	M
Series	Clinical trial series	C.7.3.2	U
–			

IE information entities, M mandatory, U user option

represent it, by limiting strings to those which are matched by a specific pattern:

```
<xs:simpleType name="AS">
  <xs:restriction base="xs:string">
    <xs:pattern value="[0-9]{3}[DWMY]"/>
  </xs:restriction>
</xs:simpleType>
```

Another example is “IS” (Integer String), it is defined as a string of characters representing an integer, it shall contain only the characters from “0” to “9”, with an optional leading “+” or “-”, the length of value is 12 bytes maximum.

XML Schema representing “IS”:

```
<xs:simpleType name="IS">
  <xs:restriction base="xs:string">
    <xs:maxLength value="12"/>
    <xs:pattern value="[+,-]([0-9])*"/>
  </xs:restriction>
</xs:simpleType>
```

With XML Schema, we gave a new description to VRs, thus we can use them to define the Data Element in the next part.

DICOM Data Element

To represent a single Data Element, we made Data Element name as XML tag, Data Element Tag and VR as XML attribute. To save storage space, Value Length can be omitted.

Take “Patient Sex” as an example:

The original data format defined by DICOM:

Tag	VR	Value Length	Value Field
(0010,0040)	CS	16	F

The format proposed by us:

XML Tag (containing attributes)	Value Field	XML Tag
<Patient_Sex group="0010" element="0040" VR="CS">	F	</Patient_Sex>

To design the XML Schema to describe this format, we defined an attribute group first, it was

named as “info”, including group number, element number, and VR:

```
<xs:attributeGroup name="info">
  <xs:attribute name="group" type="xs:string"/>
  <xs:attribute name="element" type="xs:string"/>
  <xs:attribute name="VR" type="xs:string"/>
</xs:attributeGroup>
```

The following expression describes the Data Element “Patient Sex” (Fig. 2). The data type of it is defined as the type “CS”, which has been defined previously, and the element also contains the attribute group “info”:

```
<xs:element name="Patient_Sex">
  <xs:complexType>
    <xs:simpleContent>
      <xs:extension base="CS">
        <xs:attributeGroup ref="info"/>
      </xs:extension>
    </xs:simpleContent>
  </xs:complexType>
</xs:element>
```

In this way, most of DICOM Data Elements can be represented. As there are still some complex elements that cannot be easily expressed with a general method, it is necessary to create a special approach for mapping them to XML. For instance, the Data Element whose VR is “PN” (Person Name) should contain five components, which in their order of occurrence are family name, given name, middle name, prefix, and suffix. The component delimiter shall be the caret “^”. Take “Patient Name” as an example, the expression of “John Morrison, Ph.D., Chief Executive Officer” is

Fig. 2. XML Schema for Patient_Sex.

like this: “Morrison ^John^^Ph.D., Chief Executive Officer” (one family name; one given name; no middle name; no prefix; two suffixes). To represent it with XML, each component can be treated as a child node, and the expression of the Data Element “Patient Name” is as follows:

```
<Patient_Name group="0010" element="0010" VR="PN">
  <family_name>Morrison</family_name>
  <given_name>John</given_name>
  <middle_name/>
  <prefix/>
  <suffix>Ph.D.</suffix>
  <suffix>ChiefExecutiveOfficer</suffix>
</Patient_Name>
```

Another case is that the Data Element whose VR is “SQ” (Sequence of Items) may have zero or more items. For instance, Data Element “Referenced Study Sequence” contains zero or more SOP Class/Instance pair(including “Referenced SOP Class UID” and “Referenced SOP Instance UID”) (Figs. 4

XML Schema for “Patient_Name” (Fig. 3) is as follows:

```
<xs:element name="family_name">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="given_name">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:maxLength value="64"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
...
<xs:element name="Patient_Name">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="family_name" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="given_name" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="middle_name" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="prefix" minOccurs="0" maxOccurs="unbounded"/>
      <xs:element ref="suffix" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="info"/>
  </xs:complexType>
</xs:element>
```

Fig. 3. XML Schema for Patient_Name.

DICOM Data Set

To represent a Data Set (referring to CT Image Module (Table 1)), we defined all Modules and IEs first, and then referred to them using the ref attribute.

Module Definition

In Module definition, we should indicate whether a Data Element that belongs to it is required or optional, so we followed the rules:

1. Type 1 and Type 2: required, using the minOccurs = "1" statement
2. Type 1C, Type 2C, and Type 3: optional, using the minOccurs = "0" statement

and 5). The following expression shows how to represent it with XML Schema:

```
<xs:element name="Class_Instance_Pair">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Referenced_SOP_Class_UID"/>
      <xs:element ref="Referenced_SOP_Instance_UID"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="Referenced_Study_Sequence">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Class_Instance_Pair" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attributeGroup ref="info"/>
  </xs:complexType>
</xs:element>
```

Fig. 4. XML Schema for Class_Instance_Pair.

The definition of Module (Fig. 6) is as follows:

```

<xs:element name="Patient_Module">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Patient_Name" minOccurs="1"/>
      <xs:element ref="Patient_ID" minOccurs="1"/>
      <xs:element ref="Issuer_of_Patient_ID" minOccurs="0"/>
      ...
    </xs:sequence>
  </xs:complexType>
</xs:element>
  
```

IE Definition

In IE definition, we also followed the rules to indicate whether a Module is required or optional:

1. M (mandatory): required, using the minOccurs = "1" statement
2. U (user option): optional, using the minOccurs = "0" statement

The definition of IE (Fig. 7) is as follows:

```

<xs:element name="Patient_IE">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Patient_Module" minOccurs="1"/>
      <xs:element ref="Clinical_Trial_Subject_Module" minOccurs="0"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
  
```

Data Set Definition

The definition of Data Set (Fig. 8) is as follows:

```
<xs:element name="Data_set">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="Patient_IE"/>
      <xs:element ref="Study_IE"/>
      <xs:element ref="Series_IE"/>
      ...
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

With this method, we can use XML Schema to describe the DICOM data structure. An example of Data Set is as follows:

```
<Data_Set>
  <Patient_IE>
    <Patient_Module>
      <Patient_Name group="0010" element="0010" VR="PN">
        <family_name>Morrison</family_name>
        <given_name>John</given_name>
        <middle_name/>
        <prefix/>
        <suffix>Ph.D.</suffix>
        <suffix>Chief Executive Officer</suffix>
      </Patient_Name>
      <Patient_ID group="0010" element="0020" VR="LO">000144594</Patient_ID>
      <Patient_Sex group="0010" element="0040" VR="CS">F</Patient_Sex>
      <Patient_Age group="0010" element="1010" VR="AS">35</Patient_Age>
    </Patient_Module>
  </Patient_IE>
  <Study_IE/>
  <Series_IE/>
  <Frame_of_Reference_IE/>
  <Equipment_IE>
    <General_Equipment_Module>
      <Manufacturer group="0008" element="0070" VR="LO">GE MEDICAL
SYSTEMS</Manufacturer>
      <Institution_Name group="0008" element="0080" VR="LO">Rui Jin
Hospital</Institution_Name>
      <Station_Name group="0008" element="1010"
VR="SH">RP25OC0</Station_Name>
      <Version>3</Version>
    </General_Equipment_Module>
  </Equipment_IE>
  <Image_IE/>
</Data_Set>
```

Fig. 5. XML Schema for Referenced_Study_Sequence.

Fig. 6. XML Schema for Patient_Module.

Fig. 7. XML Schema for Patient_IE.

Image Data

To store the image data, we propose JPEG format, since it belongs to the storage format supported by DICOM. The images can be encoded in base64 and treated as the content of an XML node. As a result, a series of images such as CT can be stored without repeated elements.

XSLT

XSLT is developed by W3C. It is used to transform an XML document into another XML document, or any type of document that is recognized by a browser, like HTML and XHTML. With XSLT, we can add or remove elements and attributes to or from the output file. We can also rearrange and sort elements or make decisions to hide or display certain elements. In sum, while people should originally need special software to read a DICOM file, XSLT makes it possible that the access to DICOM files can be directly obtained by a normal browser (as is shown in Fig. 9), and this could greatly facilitate the sharing of medical images.

RESULT

The XML-based DICOM data format possesses better readability and structure. Meanwhile, we can add or remove some elements whenever needed, the extensibility of DICOM files was also improved. In addition, the application of XSLT and XQuery will be demonstrated to show more benefits resulted from this new format.

XQuery

XQuery is also a W3C recommendation, it is designed to query XML data. XQuery uses XPath

Fig. 8. XML Schema for Data_Set.

expression syntax to address specific parts of an XML document. The following example shows how to query the data shown in Section “[Data Set Definition](#)”, selecting the patient’s ID whose given name is “John”:

```
for $x in doc("test.xml")/Data_Set/Patient_IE/Patient_Module
where $x/Patient_Name/given_name="John"
return <result>{$x/Patient_ID/text()}</result>
```

The result will be: “<result>000144594</result>”.

The above example suggests that the XPath expression can directly show the path to the target

element. While in DICOM, we have to scan all the data elements in their order of occurrence and lots of time would be taken in looking up Data Dictionary.

Fig. 9. Display of DICOM files on browser.

DISCUSSION

This research presented an initial approach for mapping DICOM to XML to improve the readability and the structure of DICOM files. However, there are some problems remained to be solved. For example, it is not easy to use a general method to express some complex constraint structures in DICOM, such as the multi-dimensional nuclear medicine multi-frame objects. Further work will focus on performing such mapping. Besides, since this is only a theory-based trial, certain unforeseen problems may still be encountered in practical application. More efforts will be paid to make further perfections of such content and it is expected that, in the future, most of DICOM forms can be represented with XML.

CONCLUSION

As a standard in medical imaging, DICOM is playing a more and more important role. In this

research, we proposed an XML-based DICOM data format to enhance the readability and improve the structure of DICOM files. We used XML Schema to describe the new format, making it in accordance with the one originally defined by DICOM. In addition, we have presented some examples about the technology of XSLT and XQuery, which clearly demonstrated the benefits brought from the new format, such as easily getting access to DICOM files and improved searching efficiency. Meanwhile, the improvement is favorable to further exchanging and sharing the information of medical images, making it well adapted to the Web.

REFERENCES

1. DICOM 2007, <ftp://medical.nema.org/medical/dicom>
2. You Xiaozhen, Yao Zhihong, *Application of XML in DICOM*, SPIE Volume 5748, Medical Imaging 2005, April 2005, pp. 446–454
3. Pan MinHui, Yao Zhihong, *the Application of Multilayer Tree Structure in DICOM*, SPIE Volume 6145, Medical Imaging 2006, pp. 318–328