A Container File with Relations for Cytometry and Pathology (191)

Robert C. Leif[1], Stephanie H. Leif[1]

[1]XML_Med, a Division of Newport Instruments, San Diego, USA

[1]Newport Instruments, 3345 Hopi Place, San Diego, CA, USA 92117-3516

rleif@rleif.com

A PDF of this poster is available at www.cytometryml.org

R&D, Newport Instruments, San Diego, CA, United States

Abstract

Introduction: The development of cytometry standards is complicated by the fact that much of their information space is relevant to other disciplines: medical informatics, specifically that pertaining to pathology, and biological science in general. Presently, all three groups have their own standards. Another complication is that both the objects and their relationships need to be described. CytometryML, the cytometry markup language, is an attempt to create a continuum of interoperable XML standards for flow and image cytometry that can be used by all three groups.

Methods: Wherever possible, CytometryML is based on existing standards, specifically those of the International Society for Advancement of Cytometry, ISAC, Digital Imaging and Communication in Medicine, DICOM, and International Digital Publishing Forum, IDPF. Methods: The CytometryML schemas are written in the XML Schema Definition (XSD1.1) language and validated to demonstrate adherence to the XML Schema Definition language, XSD1.1. Object oriented methodology was employed to create the CytometryML schemas. Their content was tested by translating specific XSD elements into XML and filling in the values of the objects contained therein. The attribute based syntax description of relationships in the Resource Description Framework (RDF) has been replaced and simplified by an XSD element based implementation, which permits the use of data structures. The ISAC Archival Cytometry Standard concept of a zipped data container file was further refined to be an IDPF EPUB (electronic publication) file. Since the ToC XML page locations are already present in the EPUB container, this redundant information was minimized in the Relations schema family, which replaced and extended the Table of Contents (ToC) schema of the Archival Cytometry Standard (ACS). The Relations schema includes a modified and extended version of the ToC RDF capabilities.

Results: An XML based system that includes the DICOM specified separation of series and instances and includes relationships has been created. The EPUB container file design is consistent with client-server architecture of DICOM and with addition of binary containing data files can be used as an ISAC ACS container file. The use of data structures based upon elements to describe relationships permits bidirectional and multiple relationships between two objects to be expressed. Very preliminary data indicates that the use of XSD 1.1 permits the CytometryML XML data elements to be used with XHTML5 formatting elements: <p>,<h1>, <h2>, etc. which would eventually permit the creation of a medical informatics system that has access to the full power of the Internet. A robust solution to the problem of using enumerated types where the enumerates are not expected to include all of the possibilities has been developed.

Conclusions: This DICOM based design together with the use of an EPUB container of CytometryML could serve as a reliable efficient means for the transmission of research and medical data, an extension of the pathology part of DICOM and as a prototype of an XML version of DICOM. The present implementation of a version of RDF in XSD could be extended to provide XSD1.1 with full RDF capabilities and greater functionality.

Keywords: CytometryML, Cytometry, DICOM, EPUB, FCS, Instance, Series, Schema, XML, XSD, RDF

Introduction

STANDARDS PAUCITY

- The most cost effective way to design a standard is to reuse an existing one.
- Change the syntax not the semantics!
- Benefit from the existing design, data structures, experience, semantics and documentation.
- In the case of software, design can cost more than coding. Maintenance can cost more than both.
- Evolution not revolution.
- Translate and Extend DICOM and Flow Cytometry Standard with XML schema.

Series & Instances Organization and Database Interactions

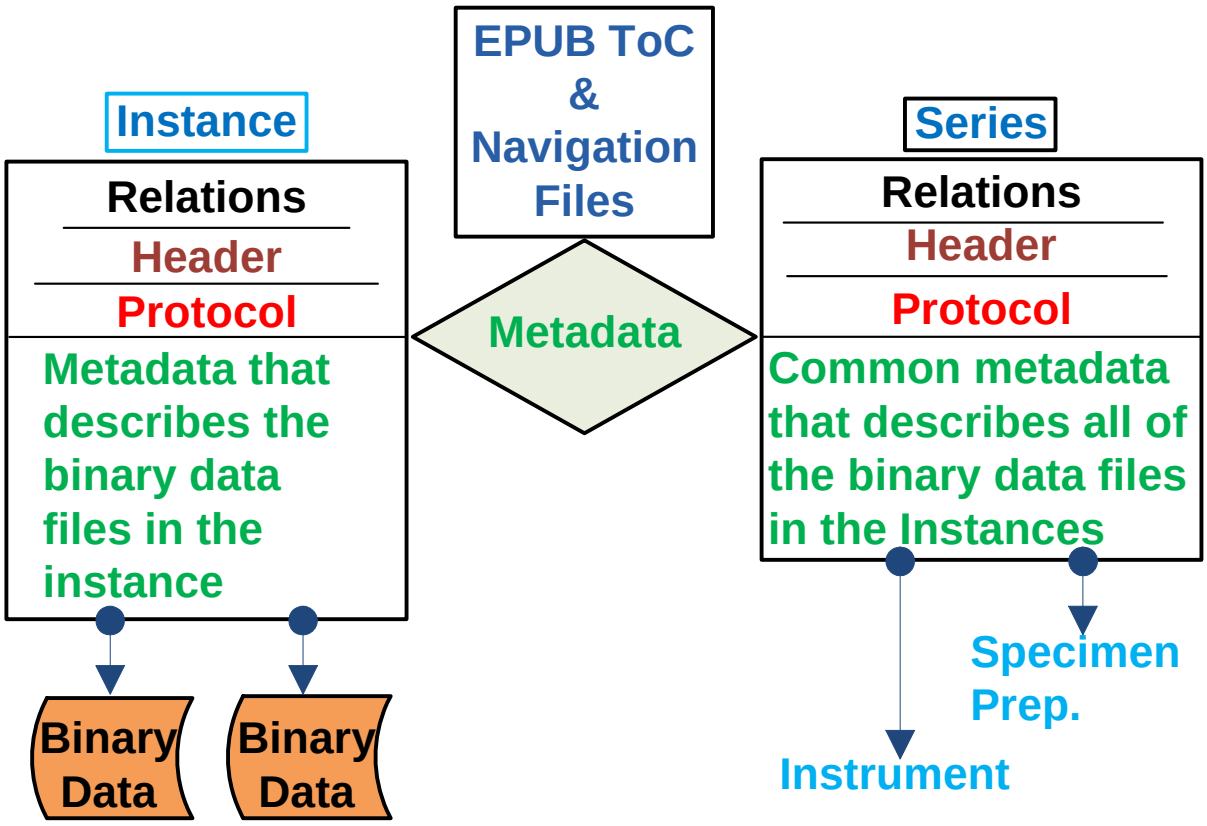

Figure 1 is a diagram showing the division of measurement metadata into the Instance and Series XML files together. with the EPUB ToC and Navigation files. The Instance_Data_Type (left) and Series_Data_Type (right) and their corresponding elements each contain Header Information, a list of relations, Relation_List, and a description of the Protocol that contains the metadata necessary to analyse the data and eventually to repeat the measurement. There can be only one Series and will often be multiple Instances.

Two URIs to the binary data are included in the metadata. These URIs permit the selected binary data to be subsequently retrieved. For DICOM, the binary files are external to the EPUB container; whereas, in the ACS, the binary files are inside of the EPUB container.

ORGANIZATION & HIERARCHY

- All of the metadata from the series together with all of the instances' metadata are retrieved together.
  --The metadata describing specific objects is retrieved on the client
- Binary or mixed data are subsequently retrieved based on the content of the metadata of specific objects
- All of the metadata from the series together with all of the instances' metadata are retrieved together.
  The metadata describing specific objects is retrieved on the client
- Binary or mixed data are subsequently retrieved
  based on the content of the metadata of specific objects

PATHOLOGY-CYTOMETRY DATA SEPARATION

- Specimen preparation should be separated into:
  --1. That which is common to all instances, which is stored with the series description.
  --2. That which is specific for each instance, which is stored with the instance.
- The instrument description should be separated into:
  --1. Those items that are unchanged for all of the instances, which are stored with the series description.
  --2. The settings and configuration that change between instances, which are stored with their specific instance.

Proposed Cytometry Metadata Organization

CYTOMETRYML DICHOTOMY IS BASED UPON THE FILE TYPE

- **List mode** (single dimensional array and/or **Images** (multidimensional arrays)
- Vector array elements in both are similar
- Organization should **not** be based on the cell transport mechanism of the Flow Cytometer or Digital Microscope!
- There is a large degree of commonality between image and flow cytometers. The only major difference is the means of specimen movement, which is only of modest significance.

Data Transfer

- ACS is a file transfer standard. DICOM is client-server standard.
- EPUB and ACS both use ZIP file containers. EPUB is a maintained widely used ZIP file based standard, which is of great importance to the book publishing business and can be used as an open standard for office type products including those used to create papers for Cytometry.
- DICOM servers have begun to use and will use Representational State Transfer (REST) RESTful Web Services:
  --Supplement 161: Web Access to DICOM Persistent Objects by RESTful Services (WADO-RS)
  --Supplement 163: STore Over the Web by RESTful Services (STOW-RS)
  --Supplement XXX: Query based on ID for DICOM Objects by RESTful Services (QIDO-RS)
- WADO Retrieve DICOM
- (studies, series, or instances by UID)
- Retrieve all metadata in one **XML** set
  --This includes ACS and/or CytometryML. **ISAC software vendors can now integrate their**

products with the Pathology Picture Archiving System (PACS)

- Retrieve bulk data (including pixels) in one multi-part MIME message

--This includes FCS files.

DICOM in XML Dilemma

- XML Schema Definition Language (XSD) is required for description of objects, such as in databases.
- Resource Description Framework (RDF) required for description of relations between objects.
- XSD and RDF have different schema languages!
- XHTML5 (W3C) presently cannot be usefully interfaced with XML. This is referred to as Polyglot Markup

SOLUTION TO XML DILEMMA

- Switch from XSD1.0 to XSD1.1
- Replace Schematron with XSD1.1 assertions
- Replace RDF with an XSD equivalent (See below).
- Requirement: optimize readability and flexibility.
- RDF Triple becomes: Subject, Verb, and Object

  --Old Idea

- Creation of the data structures required replacement of inflexible attributes with elements.

  --This increased verbosity, but provided the full power of data structures.

- Replaced loosely typed strings with strongly typed enumerations.

  --Problem: enumerations often do not include all of the needed values.

  --Solution: Use a choice element with one of the choices being essentially a string.

Since XSD1.1 includes both extension and restriction, it is an object-oriented language. Three child data-types in separate schemas have been created by restriction from Relation_Type, which is a template (generic type). These 3 instantiations are Relation_Image_Type, Relation_List_Mode_Type, and Relation_Meta_Type. Relation_Type, contains elements that are described as being anySimpleType. The instantiated elements have standard predefined data-types, which if they have a multiplicity greater than 0 and are not part of a choice element must can be included in the child schemas. Each of the anySimpleType data types becomes its own enumerated type.

Relation_Type (top of Figure 2, below) starts with 3 attributes the mime_type associated with DICOM Part 18: Web Access to DICOM Persistent Objects (WADO) with the value “application/dicom”. The second is a standard XML ID. The third attribute is a DICOM UID Value, which is a special number containing 3 to 64 character string.

Restriction of an XSD1.1 Schema

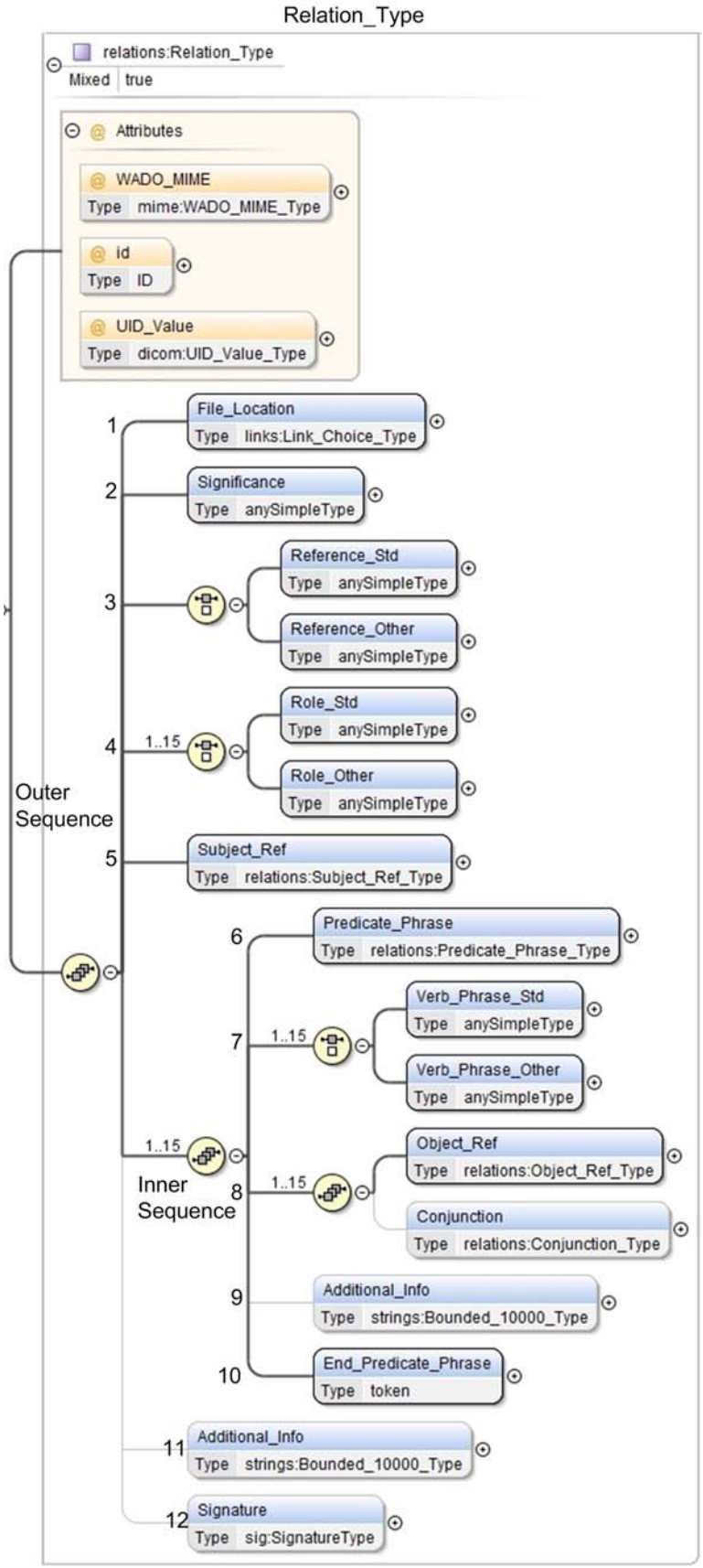

Figure 2 is a graphical description of the Relation_Type of the Restriction.xsd generic schema. The elements are numbered 1 to12. The 3 attributes are shown at the top of the figure.

Although use of strings for free text entries is easy and extremely flexible, the use of standard strings has been avoided in CytometryML except for the Additional_Info elements, which have the purpose of providing unanticipated information. Structured and constrained information greatly facilitates data analysis. For instance, manufacturer names can either be misspelled or be abbreviated. Punctuation can also add to the confusion. The simplest solution to this problem is to use enumerations; however, enumerated lists can either start incomplete or rapidly become incomplete. A solution to this problem is to provide a choice (Code Fragment 1) between a strongly typed enumeration, which should cover most responses and an Other element that is sufficiently weakly typed to accept values that were not included in the enumeration. This is demonstrated in the 3 choice elements (3, 4, 7) which each contain two anySimpleType elements. One of the 2 elements of each pair has a name that includes _Other. Upon being instantiated, the anySimpleType is replaced by an enumeration. The elements, with names that include _Other usually are replaced by a token, which is a subtype of string. This design both provides the benefits of strong typing and facilitates the use of manufacturer or user specified elements.

Code Fragment 1. Relation Choice Element

```
<choice minOccurs="1" maxOccurs="15">
   <element name="Verb_Phrase_Std" type="anySimpleType"/>
   <element name="Verb_Phrase_Other" type="anySimpleType"/>
</choice>
```

Code Fragment 1 is a choice element (7) that provides either a **Verb_Phrase_Std,** which is a predefined enumeration or **Verb_Phrase_Other**, which although it can be **anySimpleType** it is contained in a **different (Verb_Phrase_Other) element that has obviously neither been created by the software manufacturer nor is the responsibility of the software manufacturer.** The manufacture, of course will have to abjure all responsibility for other elements in the sales contract with the customer.

Code Fragment 2. Instantiation of a generic element that contains the Choice element

```
<element name="Verb_Phrase_Std"
   type="relations_image:Verb_Phrase_Image_Union_Type"
   targetNamespace="http://www.cytometryml.org/ACS/relations"/>
```

Code Fragment 2 is the instantiation of Code Fragment 1 that is present in the relations_image schema. XSD1.1 requires that the **targetNamespace** attribute be included, which aliases the namespace of the element to be the same of as in its generic parent. Note: Since the attributes in a schema do not have a namespace they neither need nor will work with the use of a **targetNamespace** in their instantiation.

Code Fragment 3 Creation of a Union_Type

```
<simpleType name="Verb_Phrase_Image_Union_Type">
    <union memberTypes="relations:Verb_Phrase_File_Type
      relations_image:Verb_Phrase_Image_Data_Std_Type"/>
  </simpleType>
```

Code Fragment 3 states that a new enumerated type has been created from one present in the relations.xsd schema and a second one present in the relations_image.xsd schema. The general members of the enumerated types mostly should be present in the parent generic schema and the specialized members in the instantiated child schemas. This approach simplifies and facilitates the maintenance of enumerated types.

RELATION_TYPE DESIGN

As is shown in Figure 2, Relation_Type includes an Outer Sequence, which includes Elements 1 to 10 and an Inner Sequence, which includes elements 6 to 10.

Element 1 in the Outer Sequence in Relation_Type is File_Location, which provides a choice of a standard link, or a hyperlink that includes a prefix and/or an IDREF link. The IDREF serves as an intra-document type of hyperlink. This permits the XML or XHTML use of relation elements to be either in stand-alone documents or as part of a complex document.

Element 2 is Significance, which in all three child schemas, presently is either “Diagnostic”, “Informational”, For “Completeness” or “Control”. In most cases, it is expected that a clinician would only look at Diagnostic Relations.

Element 3 is the Reference Choice which presently includes for both the Reference_Image and the Reference_List_Mode relations Most_Relevant_Image_Reference, Image_Reference, or Metadata Reference and for the Metadata relation List_Mode_Reference, Image_Reference, or Metadata_Reference. Since the actual data is binary, their XML description elements are references.

Element 4 is the Role Choice for which each of the 3 child schemas instantiates with its own Union element. For instance the values of the from Data_Role_File_Type are: "is Compensated Binary Data", “Contains Binary Data”, "is a One Omitted Control”, "is the Data Of Greatest Interest”, "is Data Used To Classify”, "is De-identified”, "is an Index”, "is an Instance”, "is Original_Data”, "is a Parameter_Variant”, "is Processed_Data", "is a Replicate", and is a Series Member”. and the value from the Role_List_Mode_Std_Simple_Type is “is blank without sample”.

Starting with element 5, the structure of a simple sentence is described starting with the Subject_Ref (reference), which usually is described by the word Self and thus refers to the File_Location in Element 1. If necessary, a hyperlink to a different File_Location can be included.

Element 6 is the beginning of the Inner Sequence that contains the rest of the sentence (Predicate_Phrase). Presently 1 to 15 Predicate_Phrases are permitted. This should be sufficient to describe all of the Relations of a Subject file. Since there can be only one Subject_Ref, it can not be part of the Inner Sequence.

The Relation complexType is based on the unambiguous structure of a simple sentence with four classes of elements: Subject, Verb_Phrase, Object, and Conjunction elements. The type of data structure that includes, subject. predicate (Verb_Phrase) and object is described in the draft Resource Description Framework (RDF) Vocabulary Description Language 1.1: RDF Schema (https://dvcs.w3.org/hg/rdf/raw-file/default/rdf-schema/index.html). The design presented here employes elements, which differs from RDFa (http://www.w3.org/TR/2012/NOTE-rdfa-primer-20120607/) which describes its values in the form of attributes including URIs. The Subject, Verb_Phrase, Object and Conjunction elements are ordered, since they are contained inside a sequence element; whereas, no specific grouping of RDF elements is required. As demonstrated above, the use of elements permits: inclusion of them in Choice elements, permits more than 1 of each type of element to be specified, and the use of sophisticated data structures, which include using the same Subject as the referencing entity. The Predicate_Phrase elements are solely used to delineate the predicate phrase, which consists of Verb_Phrase, Object and Conjunction elements.

When the Subject has the usual value Self, all of the elements above it describe it. The relations of the Subject are described by the elements below the Subject_Ref. Presently, it is presumed that 15 relations should be more than sufficient. The instantiations of relation.xsd provide a detailed Object-Oriented description of the Subject. Since the Subject and Object can be binary data containing files, hypertext references to these files have bee provided.

The Verb_Phrase_Std element is part of a Choice element. The other element is Verb_Phrase_Other. Verb_Phrase_Std specifies an enumeration that contains: “is parent of”, “is ancestor of”, “is child of”, “is a descendant of”, “was compensated by”, “is classification-results of”, “is the binary data described by”, “is de-identified Version of”, “is minus 1 control for”, “was analyzed with”, and “is instance of”.

A second enumeration, which is specific to the relations_meta schema is contained in the Verb_Phrase_Metadata_Std element, which is not a union element, but includes: “is an analysis-description of”, “was used to compensate”, “describes the compensation method used to produce the binary data from”, “was used to produce the compensated binary data in”, “is a classification-description of”, “describes the gates applied to”, “is an instrumentation settings description of”, “is a project-workspace of”, “is a results-description of”, “is a sample-specimen-description of”, and “is a related-publication of”.

Please note that at this stage of the development of CytometryML relations schemas, these enumerations are expected to need augmentation and revision.

Each Predicate_Phrase can include 1 to 15 Object_Ref, which are hyperlinks either to an internal part of the same large document or an external file, such as a binary containing file, such as an FCS file or an image file or a text based file, such as an XML metadata file. In order to make a relation that contains a sequence of Object_Ref elements meaningful a Conjunction element, which extends the RDF model, has been added. The values of the Conjunction enumeration presently are: or, and, and/or, xor, and not, or not, not.

The inner sequence ends with an optional element (9) for a narrative of additional information and a ending Predicate_Phrase element(10).

Relation_Type ends with a second Additional_Info Element (11) , which is followed by an electronic signature element (12).

Results & Conclusions

- Multiple CytometryML schemas have been created.

  --These schemas will need to be revised according to WG 26 and WG 27 supplements.

- The feasibility of extending DICOM and eventually translating DICOM into the XML Schema Description language (XSD) has been established.
- The problem of the inflexibility of enumerations has been solved.
- Strong typing of enumerated types can be used as an element in a choice element together with an other element providing that the other element is effectively anyString.

  --The use of enumerated types permits a fixed vocabulary to be enforced, which greatly facilitates searching the data and should greatly reduce data entry errors.

  --Since the strongly typed element and other essentially untyped element have different names, it is apparent that the values provided by the other element were nonstandard and not the work of the manufacturer.

- The use of one or more union element to combine enumerations permits a modular approach for reusing enumerations.
- The use of elements rather than attributes for simpleTypes permits the creation of data structures and

maintaining a one to one correspondence with the equivalent DICOM elements.

--choice elements do work with elements but do not work with attributes.

- XSD1.1 has features that will facilitate the creation of a standard and improve its quality.

  --Includes assertions and supports use of Object-Oriented design

  --Includes instantiation by restriction, which produced relations_image, relations_list_mode, and relations_meta from one relations schema.

- XSD1.1 potentially could be used to interface XML Elements with XHTML5
- XSD1.1 provides the functionality to Replace Schematron and RDF.
- The EPUB format can serve as the ISAC ACS container.