A copy of CytometryML Data List With Relationships is below. The code in this paper has recently been updated and is shown directly below and is followed by the original paper.

Revised Code Fragments of CytometryML Data List With Relationships

Since standard RDF includes many URIs, which are often strings of considerable length, the structure of the code can be obscured by the presence of these URIs. This problem was solved by the creation of the Compact URI or CURIE, which is the abbreviation of Compact URI.[18] A CURIE consists of a prefix, which as shown in the code snippet below, can be declared as a standard XML namespace (xmlns) attribute. As with all xmlns prefix attributes, it is followed by a colon and a reference. As is shown below in line 1 of Code Fragment 1 , the prefix is tools. "The process of evaluating (the path) involves replacing the CURIE with a concatenation of the value represented by the prefix and the part after the colon,"[18] the reference. As is shown in line 2, the reference is Compensation12Feb2011.xml. The value of the URI is file:///ACS_Flow_Tools/12Feb2011/Compensation12Feb2011.xml.

Code Fragment 1

```
1     xmlns:tools="file:///ACS_Flow_Tools/12Feb2011/"
2 <instance:CURIE>tools:Compensation12Feb2011.xml</instance:CURIE>
```

There is a prohibition that states, "CURIEs and Safe_CURIEs MUST NOT be used as values for attributes or other content that are specified to contain only URIs, IRIs, URI-references, IRI-references, etc." [18] Since schemas, such as those that comprise CytometryML contain multiple import elements and namespace attributes, that are in the form of a URI and have a common path except for the last element, the removal of the above prohibition would remove much of the clutter that is present in the beginnings of the CytometryML and other schemas, as well as the XML pages generated from them.

Code Fragment 2

```
<?xml version="1.0" encoding="UTF-8"?>
1<toc:ToC_Item
   xmlns:toc="http://www.cytometryml.org/ACS/toc_rcl"
   xmlns:tools="file:///ACS_Flow_Tools/12Feb2011/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.cytometryml.org/ACS/instance
    instance.xsd">
2<toc:Most_Relevant_Data_File_Ref mimeType="application/vnd.isac.fcs"
 id="File2">
3  <toc:Data_File_Anchor_Element href=
      "file:///file02.fcs">file02.fcs</toc:Data_File_Anchor_Element>
4     <toc:Role>
5        <toc:Role_Std>Data_Of_Greatest_Interest</toc:Role_Std>
      </toc:Role>
6     <toc:Role>
7        <toc:Role_Std>Data_Used_To_Classify</toc:Role_Std>
```

```
      </toc:Role>
8     <toc:Additional_Info>This file has the data that will be seen by
the
         person who analyzes the data</toc:Additional_Info>
</toc:Most_Relevant_Data_File_Ref>
```

Code Fragment 2 is the beginning part of a simplified example of an XML page based on the ToC_Item element of the ToC schema, which can be imported into the Instance schema. The XML page example of this element includes descriptions of two binary containing files and two XML based metadata files. Many of the values are those used in Spidlen et al[1]. The CytometryML approach differs from Spidlen et al. by being a description that is limited to a single Instance, which includes a ToC element that consists of an doubly linked list of ToC_Item elements. The order of this list is based on starting with the element that contains the description of the binary data of greatest interest to the target user.

The XML code starts with the first part of Element 1, which is the ToC_Item mentioned above. The code ends with the closure of Element 1. Element 1 starts with a list of XML namespace attributes (xmlns:) and their values, such as one that permits elements to be imported from the ToC schema, `xmlns:toc="http://www.Cytometryml.org/ACS/toc_rcl"`. The tools namespace is not that of a schema; instead as described in Code Fragment 1 , it is a CURIE prefix definition.

ToC:Most_Relevant_Data_File_Ref is the name of Element 2 of Code Fragment 2 . Element 2 includes the mimeType attribute of the first FCS Data File. The types of files included are documented by the value of the mimeType attribute, which does not constrain their content, which in theory could be anything. However, the files content can be constrained by employing elements based on data-types already present in the CytometryML schemas that specifically match individual file types. This description is continued in Elements 3-8. The URI of the most relevant binary data containing file is provided as the value of the hypertext reference (href) in Element 3. This URI ends with the file's name. The number that ends the file name is sequentially incremented as the data in the file is processed. It starts with 1 for the original raw data (Code Fragment 3, Element 10) and is incremented by 1 with each processing step.Thus, the result of a process has a greater valued number (Element 3) than that of the original raw data (Code Fragment 3 , Element 10). The data structure for Element 3 is based on an element in the XLink specification[21]. Activation of this hypertext reference opens the data file and similar links open the metadata files described in Code Fragment 4. Element 5 indicates that this file has the role of containing the binary data that was expected to be of greatest interest to the user and Element 7 indicates that this data file also has the role of being used to classify the cells. The addition of Role elements, which are based on the Role_Type in the ToC eliminates some of the need for relationships. Specifying a unique Role for a binary data containing file or XML page provides an implicit relationship (description) of itself. For example, the Role, Data_Of_Greatest_Interest, means that the Data_File referenced by this element is of greater interest than all of the other Data_Files. In order to permit Roles other than those enumerated in the standard, a choice between toc:Role_Std_Type and xlink:roleType[21] is permitted. Element 8 provides a place for a free text input. The Role and Additional_Info elements are optional and have been included in this and the other three file descriptions. An optional Signature element is also available but because of its size has not been included in Code Fragment 2, and those that follow. The inclusion of either a classic URI or CURIE equivalent (Element 3) is mandatory.

Code Fragment 3

```
9<toc:Data_File_W_Relationship_Ref mimeType="application/vnd.isac.fcs"
      id="File1">
10    <toc:Data_File_Anchor_Element href="
         file:///file01.fcs">file01.fcs</toc:Data_File_Anchor_Element>
11    <toc:Role>
```

```
12        <toc:Role_Std>Original_Data</toc:Role_Std>
13     </toc:Role>
14     <toc:Relationship  Subject="File1" Object="File2">
15        <toc:Predicate_Std>is ancestor of</toc:Predicate_Std>
          <!--File1 is the ancestor of File2.-->
       </toc:Relationship>
16     <toc:Relationship Subject="File2" Object="File1">
17        <toc:Predicate_Std>is child of</toc:Predicate_Std>
          <!-- File2 is the child of File 1  -->
       </toc:Relationship>
18     <toc:Additional_Info>This file contains the data collected by the
          flow cytometer. The cells were stained with Mouse B6.Cg 1234;
PB,
          C4,      CD8, CD16/CD32</toc:Additional_Info>
</toc:Data_File_W_Relationship_Ref>
```

Code Fragment 3 describes a second binary data containing file (Element 10), which is also an FCS file and is described in Elements 9-18. The reason for the inclusion of this Data_File_W_Relationship_Ref (Element 9) is that it's Role (Element 12) is being the original raw data that was obtained by the instrument. The number that ends the file name is lower since Most_Relevant_Data_File_Ref, which is described in elements 2-8, is a child (Element 17) of the Data_File_W_Relationship (Elements 9-18). Since this description (Data_File_W_Relationship_Ref) is of a file that is associated with another file two optional Relationship descriptions, elements 14 and 16, have been added. This addition permits a RDF type relationship between the Data_File_W_Relationship and the Most_Relevant_Data_File. It differs by being based on an element, which is derived from a complexType instead of being an attribute, which is based on a simple Type. The Relationship_Type complexType is based on the unambiguous structure of a simple sentence. This complexType consists of two attributes, Subject and Object, as well as one element Predicate. As is the case for the Role element there is a choice of the enumeration in Predicate_Std_Type or user defined values in Predicate_Other_Type. In the case of Element 14, the Subject is "File1" the Predicate (Element 15) is "is ancestor of" and the Object is "File2". This unambiguously states that File1 is the is ancestor of File2. File1 and File2 of Element 14 are both attributes of the IDREF type, which reference uniquely valued attributes of the ID type, File1 (Element 9)and File2 (Element 2). A second relationship is shown in Element 16. In this case the Subject is "File2"; the Predicate (Element 17) is "is child of" and the Object is "File1". These two relationships illustrate that this simple sentence format permits relationships between two elements that can be in both directions and that it is possible to have multiple relationships in a Data_File_W_Relationship_Ref. This is a significant difference from the unidirectionality and limitation to one relationship and one direction in RDF. The combination of the two predicates "is ancestor of" and "is child of" creates a doubly linked-list data structure, which can serve as the conceptual basis for other schemas that describe staining or specimen preparation.

Code Fragment 4

```
19<toc:Metadata_File_W_Relationship_Ref mimeType="text/xml"
   id="Compensation">
20 <toc:CURIE>tools:Compensation12Dec2010.xml</toc:CURIE>
21 <toc:Relationship Subject="Compensation" Object="File2">
22    <toc:Predicate_Std> is compensation-description of
      </toc:Predicate_Std>
```

```
        <!--Compensation is the compensation_description of File2-->
   </toc:Relationship>
23 <toc:Additional_Info>Includes compensation matrix
   </toc:Additional_Info>
 </toc:Metadata_File_W_Relationship_Ref>

24<toc:Metadata_File_W_Relationship_Ref mimeType="text/xml"
    id="Classification">
25  <toc:CURIE>tools:Classification.xml</toc:CURIE>
26    <toc:Relationship Subject="Classification" Object="File1">
27       <toc:Predicate_Std>is classification-description of
         </toc:Predicate_Std>
     </toc:Relationship>
28    <toc:Additional_Info>Includes classification algorithm
29    </toc:Additional_Info>
   </toc:Metadata_File_W_Relationship_Ref>
</toc:ToC_Item>
```

In the subsequent Relationships, shown in the two Associated_Metadata_Files, The value of the mimeType attribute (Elements 19 and 24) for both files is text/xml. The URIs for both metadata files are expressed as CURIEs (Elements 20 and 25).The values of the Subject attribute (Elements 21 and 26) are respectively Compensation and Classification. The values for the Object attribute are respectively File2 and File1 and the values of the Predicates are respectively “is compensation-description of “ (Element 22) and “is classification-description of “ (Element 27).

A CytometryML Table of Contents that Describes Relationships between Elements based upon DICOM and Flow Cytometry Standard

Robert C. Leif*[a] and Stephanie H. Leif[a]

[a]XML_Med, a Division of Newport Instruments,
3345 Hopi Place, San Diego, CA 92117
rleif@rleif.com;
http://www.newportinstruments.com

ABSTRACT

CytometryML is an XML schema based translation, extension and amalgamation of the DICOM and ISAC standards. CytometryML consists of 4 major XML schemas: Series, Instance, Instrument, and Specimen; it also includes Image and List-Mode schemas. Series metadata, which is specific for an entire collection of images and/or list-mode files produced by a single instrument and derived from a single specimen, is stored together with associated metadata files in a container (ZIP) file. Each Instance container file includes binary image and/or list-mode files together with associated metadata files that are specific for a single or closely related group of instrument runs from a single specimen. The Archival Cytometry Standard (ACS) proposed Table of Contents schema including its Resource Description Framework (RDF) capabilities has been extended and modified for use in the Instance schema.

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

1. INTRODUCTION

The International Society for Advancement of Cytometry (ISAC) Data Standards Task Force (DSTF) is now finishing work on "the Archival Cytometry Standard (ACS), which has been developed to bundle data with different components that describe cytometry experiments"[1]. The ACS captures relations among data and other components and includes support for audit trails, versioning and digital signatures. The ACS already includes Gating-ML[2], which describes gating and color compensation. The ISAC ACS container ZIP file archives the binary data and associated metadata files obtained from one or more cytometry data acquisitions of list-mode, image or similar data.

The ACS file includes an XML-based Table of Contents, which specifies file locations and relations among files in the ACS container File. Many of these files include an XML Signature, which is a W3C Recommendation[3],which has been adopted by the DSTF to allow for digital signatures of data and other components within the ACS container[1]. The present design[1] of the ACS Table of Contents (ToC) schema includes some of the functionality of the Resource Description Framework (RDF)[4,5] to document relationships.

Although the data in this archive will be of great use to many scientists, the archival format is not of direct use for the final user of clinical, routine biological and environmental measurements or applications. Transfer of a single ACS container, which can include multiple large list-mode or image files, can be slow (particularly if the Internet is involved) and provide a significant load on the data-server(s) or picture archiving and communication system (PACS). The group of measurements in the container file is analogous to the sum of the contents of a DICOM Series together with the Instance files of that Series.

The ISAC DSTF ToC files[1] employ a tree structure (Figure 1) with root element at the apex (ToC). which contains a sequence of elements based on a file element. An ACS Table of Contents file element starts with an attribute that specifies

the URI of a file and includes one associated element. This associated element includes two required attributes: one references the target of the association as a URI and the second attribute describes the type of the relationship. The relationship is that of the target of the association with the file first specified in the file element. Since both the target of the association and its relationship are attributes and only zero or one value of an attribute is permitted in XML schema, this file element can only describe one relationship.

There is a question as to which files should be the Associated_Files. They could be either the metadata files or the binary data containing files. In principle, each one of these Associated_Files could have its own Associated_Files. This can result in a complex tree of trees structure that contains multiple relationships between file descriptions located above or below a file description. In the ISAC model, since this connection is unidirectional between the file and its associated file, it would be possible to have the converse where the connection is between the associated file and the file.

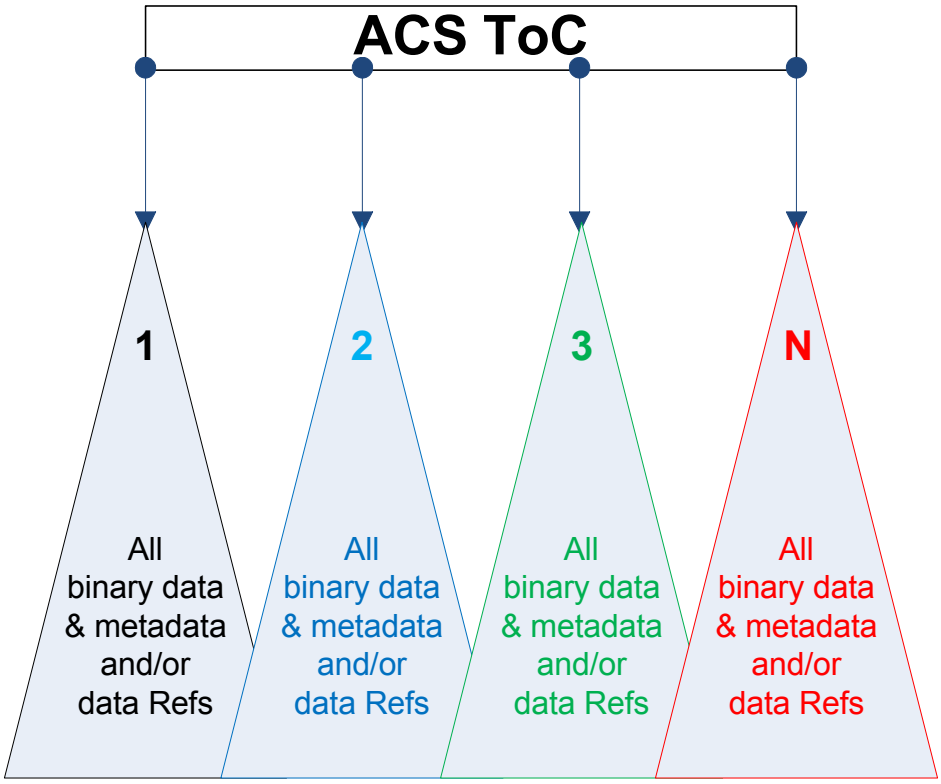

Figure 1, Structure of an ISAC Archival Cytometry Standard (ACS) table of contents. One or more XML ToC files is located at the apex of the ACS ZIP file. In this case, N ToC files are shown. The triangles represent references to trees of file descriptions. Each of these file description trees can consist of multiple file description trees.

As previously described[6], the Instance and Series schemas are based on the Digital Imaging and Communications in Medicine (DICOM) information object definitions (IODs)[7]. The Instance and Series schemas now import elements from a new Table of Contents (ToC) schema, which is to some extent based on the ACS ToC schema[1]. This paper will describe the new ToC schema, which is imported into new elements that have been added to the Instance and Series schemas. This description of the ToC element in the Instance schema is in the form of an XML page that was generated from the Instance schema. The ToC of an Instance schema can either be, as shown below, a separate XML page or be included in an Instance XML page. XML pages generated from the Instance and/or the Series schema can provide data for an XML equivalent of a DICOM structured report[8,9,10] that includes a description of pathology and/or cytometry data. A prelimi-

nary ToC element that is appropriate for a DICOM Series has been implemented, but will not be described in detail in this paper.

The use of the DICOM Series and Instance objects significantly simplifies this design by limiting the information in an Instance file container to that contained in one measurement or a closely related group of measurements. The elements located within a Series file container describe information that is relevant to all the Instances; whereas, an element located within an Instance XML page is specific for that Instance. Figure 2 is an extension of a previously described model[6]. This separation into Series and Instances limits the number of trees in any container. A Series file does not contain any binary data. In many cases, the binary data files in an Instance container file can be limited to one measurement, its controls, and processed files.

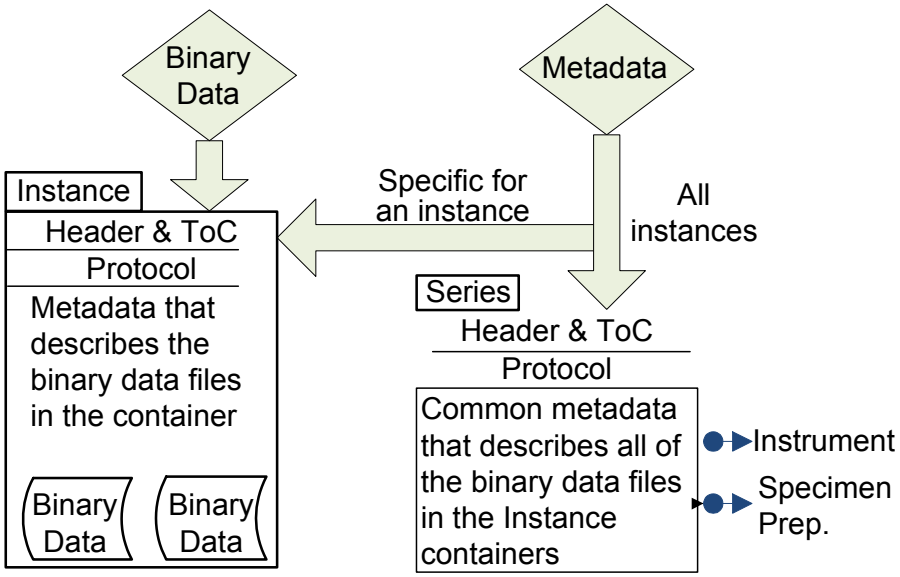

Figure 2 Diagram showing the placement of binary data and metadata into the Instance and Series containers. The Instance_Data_Type (left) and Series_Data_Type (right) and their corresponding elements each contain Header Information, a table of contents (ToC), and a description of the Protocol that contains the metadata necessary to analyse the data and eventually to repeat the measurement.

Each of the Instance container files includes a binary data containing file (top element) that is deemed to be of greatest interest to the end user. An Instance container file normally would include other binary data containing and metadata files. For example, in the case of clinical data, the top element would be the binary Data_File that was or will be used for performing the diagnosis and the Associated_Data_File could be the one that was originally produced by the instrument. In the case of research data, the top element could be the original Data_File or the file produced by compensation of the original Data_File. Each of the multiple Instance file containers can include associated binary or metadata files and their relationships. For example, the relationships could document the paths of the processing steps going back from the top element diagnostic file to the binary file that was initially produced by the cytometer.

As is shown in Figure 2 for the Series, some of the protocol information, files that contain information that is relevant to all Instance files, such as the description of the fixed components of the Instrument and all or parts of the Specimen Preparation can be resident in files that are not located in the Series container ZIP files. These separate files can be used by multiple series of measurements. In fact, the nominative versions of the Instrument file, which describes the fixed components of the instrument, and the Specimen preparation files could be maintained by the manufacturer and reside on the manufacturer's web site. No matter where these files are located, version information for them must be accessible.

2. MATERIALS AND METHODS

Much of the information and data-types present in the XML schemas and subsequently XML pages were prepared by domain experts, since the design including the element content and documentation was reused from Digital Imaging and

Communications in Medicine (DICOM)[7,8,11,12] standard (http://medical.nema.org/) or Flow Cytometry Standard, FCS3.1 (http://www.isac-net.org/images/stories/documents/Standards/fcs3.1_normativespecification_20090813.pdf). New data-types were created and data-types from other CytometryML schemas[13] were reused.

Because the Digital Imaging and Communications in Medicine (DICOM)[7] standard (http://medical.nema.org/) is a FDA Class II device[14], the safety of the software developed as part of a standard should be maximized. A strong effort was made to: maximize readability, facilitate finding the sources of elements by adherence to naming conventions, modularize the code, minimize coupling between major schemas, maximize cohesion of individual schemas, and reuse of existing CytometryML[6,15,16] XML schemas.

The syntax of the XML Schema Definition Language (XSDL) structures (http://www.w3.org/TR/xmlschema11-1/) and data-types (XSD) (http://www.w3.org/TR/xmlschema11-2/) were helpful in this attempt to produce safe code. In the future, XSDL version 1.1 also includes assertions, which provide extra checks on the correctness of the code. The requirement of readability should satisfy the requirement that a "Detailed semantic shall be provided to prevent potential misinterpretations and misusages of the standard", which has been included in a requirement of the ISAC Advanced Cytometry Standard (ACS) for a data file standard[17]and many of the other requirements are met for a data file standard or facilitated by the use of XSDL, and the structure of CytometryML. The absence of methods (functions and procedures) greatly simplified the process; however, it does not permit the CytometryML schemas to be the complete basis of any web page or other entity that is more than a passive document.

The XSDL schemas were validated by oXygen (http://www.oxygenxml.com/) and XMLSpy (http://www.altova.com/). Many of the schemas have also been tested with XSDL 1.1 with the Saxon-EE 9.2.0.3 parser. A new XML page was subsequently produced from each of the main schemas and then filled with reasonable values and validated against its schema.

3. RESULTS

A Table of Contents element has been added to the ToC, Series and Instance schemas and prototyped for the ISAC standard ACS Container file. Effectively, the ACS Container file design has been simplified by following the DICOM design, which divides the content of the ACS container into a Series and Instances hierarchy.

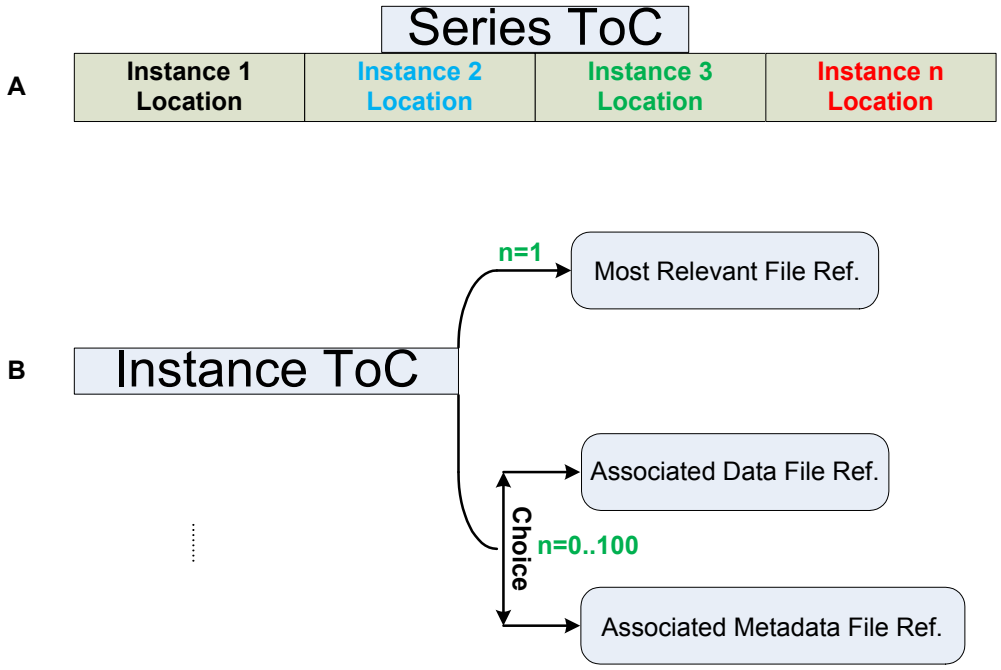

Figure 3, CytometryML Design of ToC elements. As shown at the top of the figure (A), the Series ToC consists of a list of the locations of the Instance files. As shown in the middle and bottom of the figure (B) and described below, an Instance ToC is a significantly more complex structure. Both binary and metadata files can be associated with another file.

There has been considerable interest and work on the creation of Semantic Webs of data (http://www.w3.org/standards/semantic web/). According to World Wide Web Consortium (W3C), "The Semantic Web is a Web of data — of dates and titles and part numbers and chemical properties and any other data one might conceive of. The Resource Description Framework (RDF) (http://www.w3.org/RDF/) provides the foundation for publishing and linking your data." RDF is a way of documenting the relationships between objects. These objects can be represented by URIs. Most embodiments of RDF technology employ schemas that are different and often incompatible with the XML Schema Definition Language (XSDL), which is the basis of CytometryML and the ACS schemas. A very useful way to employ the equivalent of RDF in XHTML web pages is to use a special set of attributes. However, it would be useful to describe relationships employing RDF in standard XML schemas. The ToC schema of CytometryML includes a construct to describe relationships between file references. This construct is a Relationship element. The description of Relationship_Type is changed from the standard attribute of RDFa to an element. Since an attribute in a schema can only generate or validate one attribute value pair in an XML page, the change to an element, which can generate or validate unlimited element value pairs now permits more than one relationship between entities.

Since standard RDF includes many URIs, which are often strings of considerable length, the structure of the code can be obscured by the presence of these URIs. This problem was solved by the creation of the Compact URI or CURIE, which is the abbreviation of Compact URI.[18] A CURIE consists of a prefix, which as shown in the code snippet below, can be declared as a standard XML namespace (xmlns) attribute. As with all xmlns prefix attributes, it is followed by a colon and a reference. As is shown below in line 1 of Code Fragment 1 , the prefix is tools. "The process of evaluating (the path) involves replacing the CURIE with a concatenation of the value represented by the prefix and the part after the colon,"[18] the reference. As is shown in line 2, the reference is Compensation12Dec2010.xml. The value of the URI is file:///ACS_Flow_Tools/12Dec2010/Compensation12Dec2010.xml.

Code Fragment 1

```
1.     xmlns:tools="file:///ACS_Flow_Tools/12Dec2010/"
2. <instance:CURIE>tools:Compensation12Dec2010.xml</instance:CURIE>
```

There is a prohibition that states, “CURIEs and Safe_CURIEs MUST NOT be used as values for attributes or other content that are specified to contain only URIs, IRIs, URI-references, IRI-references, etc.” [18] Since schemas, such as those that comprise CytometryML contain multiple import elements and namespace attributes, that are in the form of a URI and have a common path except for the last element, the removal of the above prohibition would remove much of the clutter that is present in the beginnings of the CytometryML and other schemas as well as the XML pages generated from them.

Code Fragment 2

```
<?xml version="1.0" encoding="UTF-8"?>
1<instance:Data_File_With_Associations
   xmlns:mime="http://www.Cytometryml.org/ACS/mime"
   xmlns:strings="http://www.Cytometryml.org/ACS/strings"
   xmlns:curies="http://www.Cytometryml.org/ACS/curies"
   xmlns:instance="http://www.Cytometryml.org/ACS/instance"
   xmlns:sig="http://www.w3.org/2000/09/xmldsig#"
   xmlns:toc="http://www.Cytometryml.org/ACS/toc_rcl"
   xmlns:tools="file:///ACS_Flow_Tools/12Dec2010/"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.Cytometryml.org/ACS/instance
      instance.xsd">
2  <instance:Data_File_Ref toc:mimeType="application/vnd.isac.fcs">
3     <toc:Data_File_URI>file:///file02.fcs</toc:Data_File_URI>
4     <toc:Role>Data_Of_Greatest_Interest</toc:Role>
5     <toc:Role>Data_Used_To_Classify</toc:Role>
6     <toc:Additional_Info>This file has the data that will be seen by
         the person who analyzes the data</toc:Additional_Info>
   </instance:Data_File_Ref>
```

Code Fragment 2 is the beginning part of a simplified example of an XML page based on the Data_File_With_Associations element of the Instance schema. The XML page example of this element includes descriptions of two binary containing files and two XML based metadata files. Many of the values are those used in Spidlen et al[1]. The CytometryML approach differs from Spidlen et al. by being a description that is limited to a single Instance, which includes a ToC element that consists of an array of Data_File_With_Associations elements. The order of this array is based on starting with the element that contains the description of the binary data of greatest interest to the target user.

Instance:Data_File_With_Associations is the name of Element 1 of Code Fragment 2 . Element 1 starts with a list of XML namespace attributes (xmlns:) and their values, such as one that permits elements to be imported from the ToC schema, `xmlns:toc="http://www.Cytometryml.org/ACS/toc_rcl"`. The tools namespace is not that of a schema; instead as described in Code Fragment 1 , it is a CURIE prefix definition.

Element 2 includes the mimeType attribute of the first FCS Data File. The types of files included are documented by the value of the mimeType attribute, which does not constrain their content, which in theory could be anything. However, the files content can be constrained by employing elements based on data-types already present in the CytometryML schemas that specifically match individual file types. This description is continued in Elements 3-6. The URI of the most relevant binary data containing file is provided in Element 3. This URI ends with the file’s name. The number that ends the file

name is sequentially incremented as the data in the file is processed. It starts with 1 for the original raw data (Code Fragment 3, Element 8) and is incremented by 1 with each processing step.Thus, the result of a process has a greater valued number (Element 3) than that of the original raw data (Code Fragment 3 , Element 8). Element 4 indicates that this file has the role of containing the binary data that was expected to be of greatest interest to the user and Element 5 indicates that this data file also has the role of being used to classify the cells. The addition of Role elements, which are based on the Role_Type in the ToC eliminates some of the need for relationships. Specifying a unique Role for a binary data containing file or XML page provides an implicit Relationship. For example, the Role, Data_Of_Greatest_Interest, means that the Data_File referenced by this element is of greater interest than all of the other Data_Files. Element 6 provides a place for a free text input. The Role and Additional_Info elements are optional and have been included in this and the other three file descriptions. An optional Signature element is also available but because of its size has not been included in Code Fragment 2, and those that follow. The inclusion of either a classic URI or CURIE equivalent (Element 3) is mandatory.

Code Fragment 3

```
7  <instance:Associated_Data_File_Ref
       mimeType="application/vnd.isac.fcs">
8      <instance:Data_File_URI>file:///file01.fcs</instance:Data_File_URI>
9      <instance:Role>Original_Data</instance:Role>
10     <instance:Relationship
          toc:Direction="Source=Data_Target=Associated">
11        <toc:Relationship_ACS>child</toc:Relationship_ACS>
       </instance:Relationship>
12  <instance:Relationship toc:Direction="Source=Associated_Target=Data">
13        <toc:Relationship_ACS>ancestor</toc:Relationship_ACS>
14     </instance:Relationship>
15  <instance:Additional_Info>This file contains the data
          collected by the flow cytometer. The cells were stained
          with Mouse B6.Cg 1234; PB, C4, CD8, CD16/CD32
       </instance:Additional_Info>
    </instance:Associated_Data_File_Ref>
```

Code Fragment 3 describes a second binary data containing file (Element 7), which is also an FCS file and is described in Elements 8-15. The reason for the inclusion of this Associated_Data_File_Ref (Element 7) is that it's Role (Element 9) is being the original raw data that was obtained by the instrument. The number that ends the file name is lower since Data_File_Ref, which is described in elements 2-6, is a child (Element 11) of the Associated_Data_File (Elements 7-15). Since this description (Associated_Data_File_Ref) is of a file that is associated with another file two optional Relationship descriptions, elements 10 and 12, have been added. This addition permits a RDF type relationship between the Associated_Data_File and the Most_Relevant_Data_File. It differs by being based on an element, which is derived from a complexType. Previous relationship descriptions are in the form of attributes, which are based upon simpleTypes. The Relationship_Type complexType contains a Direction attribute. In the case of Element 10, the Direction is Source=Data_Target=Associated and the value is child. This unambiguously states that Data_File is the child of the second binary data containing file. A second relationship is shown in Elements 12 and 13. In this case the Direction is Source=Associated_Target=Data and the value is ancestor. This bidirectional ability and that of having multiple relationships is a significant difference from the unidirectionality and limitation to one relationship and one direction in RDF.

Code Fragment 4

```
16 <instance:Associated_Metadata_File_Ref mimeType="text/xml">
17    <instance:CURIE>tools:Compensation12Dec2010.xml</instance:CURIE>
18    <instance:Relationship
         toc:Direction="Source=Associated_Target=Data">
19       <toc:Relationship_ACS>compensation_description
         </toc:Relationship_ACS>
      </instance:Relationship>
20    <instance:Additional_Info>Includes compensation matrix
      </instance:Additional_Info>
   </instance:Associated_Metadata_File_Ref>

21 <instance:Associated_Metadata_File_Ref mimeType="text/xml">
22    <instance:CURIE>tools:Classification.xml</instance:CURIE>
23    <instance:Relationship
toc:Direction="Source=Associated_Target=Data">
24       <toc:Relationship_ACS>classification_description
      </toc:Relationship_ACS>
</instance:Relationship>
25    <instance:Additional_Info>Includes classification algorithm
      </instance:Additional_Info>
   </instance:Associated_Metadata_File_Ref>
</instance:Data_File_With_Associations>
```

In the subsequent Relationships, shown in the two Associated_Metadata_Files, the value of the Direction attribute (Elements 18 and 23) is Source=Associated_Target=Data and the values of the Relationships are respectively compensation_description (Element 19) and classification_description (Element 24). The ACS suffix indicate that the Relationship values were specified in the standard. Relationships that are not part of the standard would have Other as the suffix. The value of the mimeType attribute (Elements 16 and 21) for both files is text/xml. The URIs for both metadata files are expressed as CURIEs.

4. CONCLUSIONS

The ACS design for a Table of Contents (ToC) has been extended and modified in CytometryML to an XML schema datatype that is appropriate for the DICOM Instance data structure. The ToC has also been extended to include a Role element and Relationship elements based on an extended Relationship_Type that includes a Direction attribute, which disambiguates the RDF expression, and also permits multiple relationships with different directions between two files. This combination of Role and enhanced Relationship types should provide a much richer vocabulary to describe objects and their relations than present RDF. The CytometryML design and that of Spidlen et al.[1] have the advantage that since they have been created in XSDL, they can be easily imported into other XSDL schemas. The separation of the descriptions of and data from cytometry measurements into a Series, which primarily contains metadata that is applicable to multiple Instances, each of which contains a limited set of binary data, will decrease the amount of data transmitted. This decrease will, as experienced with DICOM, provide improved response times and lower costs.

In this work, the reuse of an ACS construct in a DICOM based data structure has been demonstrated, which is an extension of previous reuse of DICOM data structures in CytometryML. This leads to the conclusion that the ACS and CytometryML should both reuse DICOM and DICOM should reuse ACS and CytometryML. These standards should be harmonized at the datatype semantics level. The harmonization of these standards and similar work on harmonizing other medical standards and translating them into XML[13,19,20] will significantly assist in meeting Executive Office of the President's Council of Advisors on Science and Technology's goal[21] of "capability for universal data exchange." Since DICOM usually is tested by being implemented in a compiled language, the creation of XML schemas and pages presents an opportunity for early, but partial testing of a DICOM design.

5. ACKNOWLEDGMENT

This work was sponsored by Newport Instruments internal research funds. One of us, R.C.L. wishes to thank Ryan R. Brinkman, Josef Spidlen and the other members of the ISAC DSTF for many enlightening and pleasant discussions.

6. REFERENCES:

1] Spidlen, J. et al., "Archival Cytometry Standard ACS, International Society for Advancement of Cytometry, Draft Candidate Recommendation," Available at: http://flowcyt.sf.net/acs/latest.pdf (2010).

2] Spidlen, J., Leif, R.,C., Moore, W., Roederer, M., International Society for the Advancement of Cytometry Data Standards Task Force, Brinkman, R. R., "Gating-ML: XML-based gating descriptions in flow cytometry," Cytometry Part A 73A, 1151-1157 (2008).

3] World Wide Web Consortium (W3C)., "W3C Recommendation - XML Signature Syntax and Processing (Second Edition),". Available at: http://www.w3.org/TR/xmldsig-core/2010 (2010).

4] Beckett D. (Editor), "RDF/XML Syntax Specification (Revised), W3C Recommendation 10 February 2004," Available at: (http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/) (2004).

5] Adida, B., Birbeck, M., McCarron, S., and Pemberton, S. (Editors), "RDFa in XHTML: Syntax and Processing, A collection of attributes and processing rules for extending XHTML to support RDF", W3C Recommendation 14 October 2008, Available at: (http://www.w3.org/TR/rdfa-syntax/) (2008).

6] Leif, R.,C.,"Toward the integration of cytomics and medicine," J. Biophoton. 2, 482–493 (2009).

7] Pianykh O.S., [Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide], Springer Publishers, Berlin & Heidelberg, (2008).

8] "DICOM) Part 3: Information Object Definitions, Annex A.35 Structured Report Document Information Object Definitions", (p. 220) Available at: ftp://medical.nema.org/medical/dicom/2009/ (2009).

9] D. A. Clunie, "DICOM Structured Reporting", PixelMed Publishing ISBN 0-9701369-0-0 Available at: http://www.pixelmed.com/srbook.html (2005).

10] Lee, K.,P., Hu, J., "XML Schema Representation of DICOM Structured Reporting," Journal of the American Medical Informatics Association Volume 10, 213-223, http://www.ncbi.nlm.nih.gov/pmc/articles/PMC150374/pdf/0100213.pdf (2003).

11] DICOM Working Group 26, "DICOM Supplement 145: Whole Slide Imaging in Pathology," which applies to Parts 2,3,4,6,16,17 of the DICOM Standard Available at: (http://www.dclunie.com/dicom-status/status.html#BaseStandard2001) (2009)

12] DICOM Working Group 26, "DICOM Supplement 122: Specimen Module and Revised Pathology SOP Classes," applies to Parts 2,3,4,6,16 of the DICOM Standard Available at: http://www.dclunie.com/dicom-status/status.html#BaseStandard2001 (2008).

13] Leif R.C., Leif S.B., and Leif S.H., "CytometryML, an XML Format Based on DICOM and FCS for Analytical Cytology Data," Cytometry 54A, 56-65 (2003).

14] Office of Device Evaluation, "Guidance for the Submission Of Premarket Notifications for Medical Image Management Devices", US FDA, Available at: http://www.fda.gov/downloads/MedicalDevices/DeviceRegulationandGuidance/GuidanceDocuments/ucm073721.pdf (2000).

15]. Leif R.C., Spidlen J., Brinkman R.R., "A Container for the Advanced Cytometry Standard (ACS)," Proc. SPIE 7182, 71821Q (2009).

16] Leif R.C., "CytometryML, Binary Data Standards," Proc. SPIE 5699, 325-333 (2005).

17] Spidlen J., Brinkman R., Leif R. C., and other members of the ISAC Data Standards Task Force., “Advanced Cytometry Standard (ACS) Requirements for a data file standard format to describe cytometry and related analytical cytology data, Version 0.070920,” Available at: http://cdnetworks-us-1.dl.sourceforge.net/project/flowcyt/Analytical Cytology Standard/Analytical Cytology Standard/Requirements-v070920.pdf (2007).

18] Birbeck, M., and McCarron, S. (Editors) “CURIE Syntax 1.0, A syntax for expressing Compact URIs, W3C Candidate Recommendation 16 January 2009,” Available at: http://www.w3.org/TR/2009/CR-curie-20090116 (2009).

19] Catley, C., Frize, M., "Design of a health care architecture for medical data interoperability and application integration," [Engineering in Medicine and Biology, 2002. 24th Annual Conference and the Annual Fall Meeting of the Biomedical Engineering Society EMBS/BMES Conference, 2002. Proceedings of the Second Joint 3, 1952- 1953 (2002).

20] Hongli, L., Chen, Z., Wang, W., “XML Schemas Representation of DICOM Data Model,” Bioinformatics and Biomedical Engineering (iCBBE), 2010 4th International Conference,” (2010).

21] Executive Office of the President, President’s Council of Advisors on Science and Technology, “Report To The President, Realizing The Full Potential Of Health Information Technology To Improve Healthcare For Americans: The Path Forward” Available at: http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-health-it-report.pdf (2010).