A Container for the Advanced Cytometry Standard (ACS)

Robert C. Leif[1]*
Josef Spidlen[2]
Ryan R. Brinkman[2]
[1]XML_Med, a Division of Newport Instruments, 5648 Toyon Road, San Diego, CA, USA 92115-1022, rleif@rleif.com
[2]Department: Terry Fox Laboratory, BC Cancer Research Centre, 675 West 10th Avenue, Vancouver, BC Canada V5Z 1L3

ABSTRACT

Introduction: The highest priority for the Advanced Cytometry Standard (ACS) is the interpretation of list-mode cytometry measurements. Other priorities of lesser importance are the capacity to reproduce a cytometry measurement and the implementation of a digital microscopy image standard. The sequential nature of these requirements is being accommodated by a flexible, modular design. A major feature of this modular design is the creation of a design for an Advanced Cytometry Standard Container (ACSC) that includes a Table of Contents (ToC) XML file, one or more binary data containing files and files that contain the meta-data that describes the binary data.

Methods: The design and partial implementation of the CytometryML schemas have been based on the techniques of modularity (each schema describing one object), iterative (spiral) development, inheritance, and reuse. Data-types including their definitions have been reused from DICOM, FCS, and other standards.

Results: A prototype ToC schema together with prototypes of many of the schemas that describe the contents of the ACSC have been created together with their supporting schemas. These schemas have been validated with two tools and XML pages were generated from the main element(s) of the highest level schemas. These elements describe the table of contents of the zipped container file and a flow-cytometry instrument. The zipped container file (ACSC) describes and contains the meta and binary data.

Keywords: Advanced Cytometry Standard, ACSC, container, flow cytometry, DICOM, FCS, XML schema, ZIP

1. INTRODUCTION

The International Society for the Advancement of Cytometry (ISAC) Data Standards Task Force (DSTF) has been charted to develop a new data standard for cytometry. Presently, there are discussions concerning the content of the initial release of the ISAC Advanced Cytometry Standard (ACS)[1] and the organization of the ACS schemas. The design and content of the material below is an attempt to create parts of the ACS.

The ACS requirements[2] dictate a phased approach with completion of Requirement 1 (Req. 1) before Req.2. Req. 1 from the Advanced Cytometry Standard (ACS) is "to provide a mechanism to record all the essential information for a person knowledgeable in the field to interpret a cytometry experiment." And Req. 2 is "to record all the essential information to reproduce a cytometry experiment." Req. 1 is a subset of Req. 2. Much of Req. 2 has been, at least partially, fulfilled by reengineering the CytometryML schemas to provide the functionality specified by existing drafts of the proposed ACS schemas and design documents[3]. There is also the desire of the (DSTF) to develop a new standard for flow cytometry first and then to follow it with standards for image and other cytometry modalities (Req. 5).

The design of the ACS should include cognizance of three facts: 1) Conformance to all of the requirements will result in a code base that will need to be maintained and extended; 2) this maintenance will eventually be performed by others rather than the authors; and 3) the authors have limited memories. The problems of maintenance are further exacerbated by the flexibility of software, which permits and encourages useful upgrades. Upgrades, although necessary, will increase the difficulty of maintaining the software base that describes the standard.

*rleif@rleif.com; phone 1 619 582-0437; newportinstruents.com

The Advanced Cytometry Container (ACSC) is a zip file[4,5] that includes the binary data from one or more cytometry measurements, one Table of Contents (ToC) XML file, and other files describing the meta-data associated with the binary data. The design proposed in this paper for the ToC schema that is used to validate the ToC XML files includes a high level overview of the organization of the rest of the project, the contents of the ACSC.The ACS container file has been described in the ACS Container Usage Recommendation[6]. Fig. 1 has been adapted from that document.

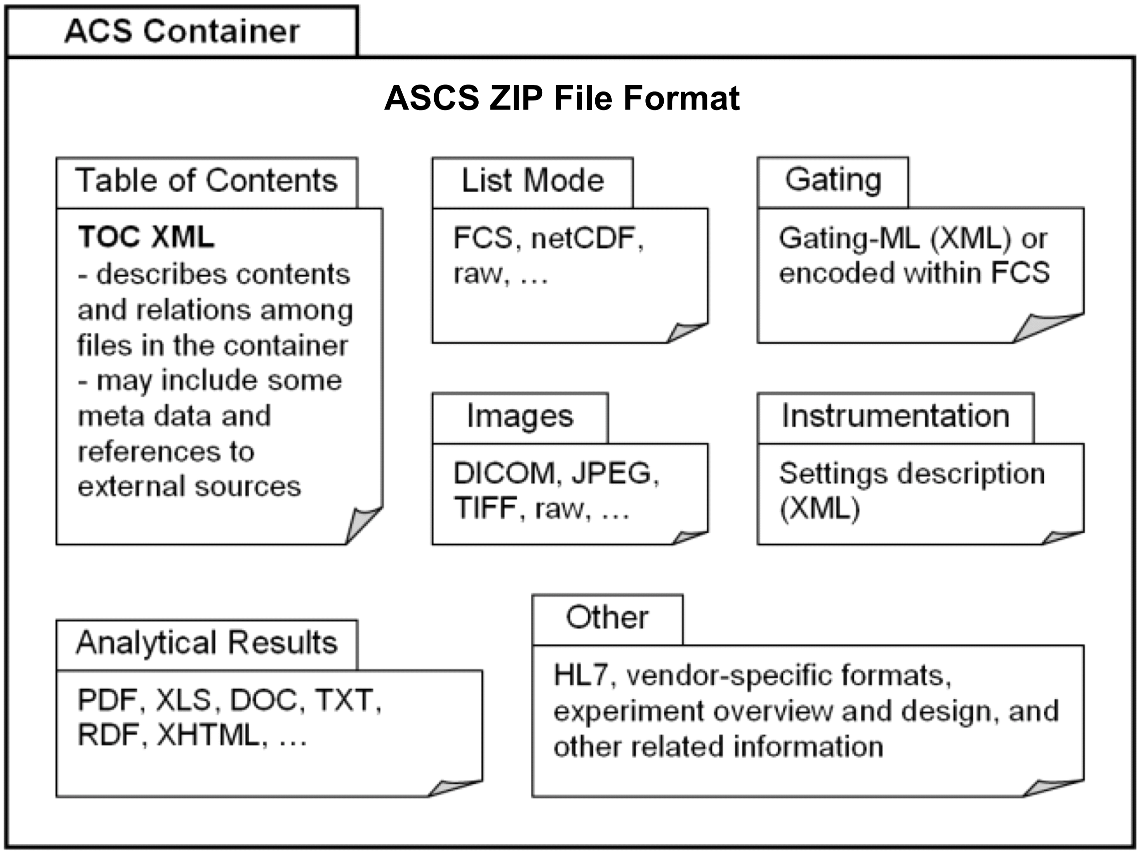
ACS Container
ASCS ZIP File Format
Table of Contents
TOC XML
- describes contents and relations among files in the container
- may include some meta data and references to external sources
List Mode
FCS, netCDF, raw, …
Gating
Gating-ML (XML) or encoded within FCS
Images
DICOM, JPEG, TIFF, raw, …
Instrumentation
Settings description (XML)
Analytical Results
PDF, XLS, DOC, TXT, RDF, XHTML, …
Other
HL7, vendor-specific formats, experiment overview and design, and other related information

Fig. 1 is a chart showing the contents of the ACS container (ACSC), which is a ZIP file. List-mode and other binary data are stored in files together with the meta-data that describes the information contained in the binary files. Other files that are also relevant to the binary data can be stored within the container.

The Table of Contents (ToC)[6], which is shown at the upper left, describes the contents of the ZIP file including universal resource identifiers (URIs)[7] that point to meta-data files that are relevant to the files contained within the ASCS ZIP file. The meta-data files can be internal or external to the ASCS file. The schema that describes the ToC will be discussed below. The pointer to a list-mode file is located at the top center. Presently three formats for list-mode have been included: flow cytometry standard (FCS)[8], Network Common Data Form (netCDF)[9,10], and raw[11] are shown. Raw is simply the binary data in a list-mode file, which is a unidimensional array of vectors in the form of records or structs of electronic parameter values[11]. Similarly as shown at the center of the figure, images can be included. The location and formats of the descriptions of the meta-data that describes the processing of the binary data are shown at the right. This includes the gating[12] (top), the description of the instrument including its settings (middle) and other files (lower). The location of the Analytical results, which can be in standard commercial file formats is shown at the bottom left.

2. METHODS

The design of the schemas was modular[13,14], iterative, and essentially followed the spiral method of development[15]. The existing CytometryML schemas were updated to conform to ISAC DSTF suggestions and consensus items. Other changes were made in accordance with the design of Digital Imaging and Communications in Medicine[16] (DICOM) working groups. A simple object based design methodology was employed; and the schema were treated as if they were the descriptions of the data-types in the classes or packages of object-oriented languages. The size of the individual schemas was kept manageable by budding off individual data-types that had become large and/or complex and their components into a new schema. Inheritance was possible by extension for data-types in both their parent schemas and other schemas that imported the parent schemas. Unfortunately, the present version of the XML schema standard does not permit restricting a complexType that is imported from one schema to another[17].

In order to maximize simplicity, the schemas were and are being constructed to be an acyclic graph. No schema was imported by any that it imports. Import diagrams was created from the individual CytometryML schemas by transferring one of them from Altova XMLSpy to Altova Schema Agent. The vector image created by the Altova Schema Agent was then stored as a Windows enhanced metafile (EMF) file. This EMF file was extensively formatted in VISIO (http://office.microsoft.com/en-us/visio/default.aspx).

Datatypes were reused[18] from the DICOM standard[16,19] and FCS[8], and numerical types were reused from ECMA-International (http://www.ecmainternational.org/). The CytometryML schemas were developed using the XML schema definition language, XSDL[20,21], and were validated with both Stylus Studio (http://www.stylusstudio.com) and XMLSpy (http://www.altova.com). These schemas are primarily derived from DICOM datatypes with XSDL documentation elements that included references to the descriptions of the datatypes in the DICOM standard[16] and datatypes that could be part of an extension of DICOM. The visibility of the source of datatypes and elements has and is being maximized by importing rather than including all schemas. A schema's xlmns attribute value including its prefix was reused when the schema was imported into other schemas. The numerous, ubiquitous xs: prefixes have been eliminated by not including a prefix in the value of the xlmns attribute for schema and instead including a prefix based on the name of the schema for the xlmns attribute that makes the schema's elements visible unto itself.

3. RESULTS

A Table of Contents (ToC) schema has been created. In accordance with the methodology described above, the ToC imports 9 schemas of which 8 are shown in Fig. 2. One of the schemas, about.xsd, was omitted because it is imported by every schema. About.xsd provides a description of the meta-data that describes each schema. This data includes the author(s) names and the maintainer's name and e-mail address, etc.

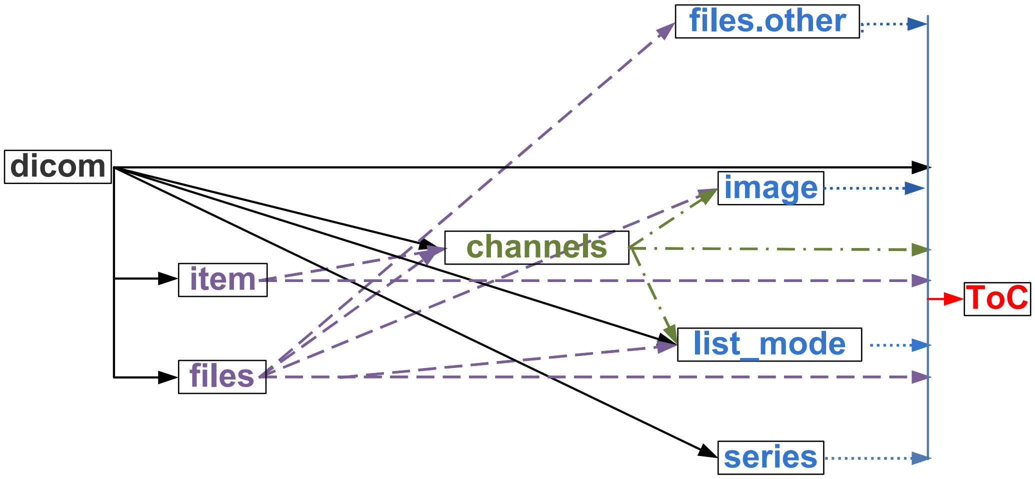

Fig. 2 is a chart that shows all of the schemas directly imported by the ToC schema except about, which is imported by all of the schemas. Schemas that are imported by other schemas have been placed to the left. For instance, even though channels is directly imported by ToC, channels has been placed to the left of image and list_mode, which also import it.

The five major schemas are channels, image, list_mode, series, and ToC. Item, dicom, files, files.other are helper schemas that supply general purpose data-types to many schemas. The imported data-types supply the elements either directly or indirectly to the ToC_Type.

Code Fragment 1 (Schema)

```
1<complexType name="ToC_Type">
2 <sequence>
3   <element name="Binary_Data_Info"type="toc:Binary_Data_Info_Type"/>
4   <element name="Series_Info" type="files:XML_File_Type"/>
5   <element name="Instrument_Data_Info" type="toc:Instrument_Data_Info_Type"/>
6   <element name="Compensation_File_Info"
      type="toc:File_Format_Modifiable_Data_Type"/>
7   <element name="Gating_Method_Info"
      type="toc:File_Format_Modifiable_Data_Type"/>
8   <element name="Analysis_Method_Info"
      type="toc:File_Format_Modifiable_Data_Type"/>
9   <element name="Experiment_Overview_Description"
      type="toc:XML_or_Other_File_Type"/>
10  <element name="Sample_Specimen_Details" type="toc:XML_or_Other_File_Type"/>
11  <element name="Result_File_Description" type="toc:XML_or_Other_File_Type"/>
12  <element name="Report_file_Description" type="toc:XML_or_Other_File_Type"/>
13  <element name="Other_File_Info"type="toc:XML_or_Other_File_Type"/>
14  <element name="Additional_Info" type="anyType" minOccurs="0"
      maxOccurs="unbounded"/>
  </sequence>
</complexType>
```

Code Fragment 1 shows the section of the Table of Contents schema (toc.xsd) that describes the ToC_Type. As shown in Element 1 (Line 1), ToC_Type is the name of a complexType. It is termed complex in XSDL because it contains multiple elements. The numbers at the beginning of a line indicate the start of a new element. Since a sequence is a type of element its first line is numbered (Element 2). A sequence is equivalent to a record or struct construct in standard programming languages. The first element in the sequence (Element 3), Binary_Data_Info, describes the location of and structure of the binary data file. The prefix, toc:, before the name of the data-type indicates that the data-type is from the toc.xsd schema. The sequence of ToC_Type contains 12 elements (Elements 3 to 14).

After a schema has been validated, the simplest way to determine if it produces the intended results is to generate an XML page from one of its elements. XMLSpy was used to generate an XML page from the ToC element. The Binary_Data_Info element part of that XML page is shown in Code Fragment 2 below. Imagined values of the elements and attributes have been used to filled in the XML page.

Code Fragment 2 (XML)

```
1<toc:Binary_Data_Info>
2   <toc:Acquisition_Context>
3      <toc:Acquisition_Number>483648</toc:Acquisition_Number>
4      <toc:Instance_Number>1</toc:Instance_Number>
5      <toc:Instance_Role>
6         <toc:Variable>
7            <toc:Variable_Value>1000</toc:Variable_Value>
8            <toc:Variable_Name>PMT Volts</toc:Variable_Name>
         </toc:Variable>
```

```
      </toc:Instance_Role>
    </toc:Acquisition_Context>
9   <toc:List_Mode_File_Info>
10  <list:FCS File_Extension="fcs">
11    <list:File_Information files:originality="Original"
        files:required_to_process="true" files:Text_Status="Normative"
        files:File_Name="483648"
        files:File_URI=”http://www.newportinstruments.com/21Nov08/483648"
        files:description="normalizedString" files:Format="FCS3.1"
        files:File_Content="Mixed">
12        <files:Instance_Creation_Date_Time>2008-11-21T09:30:47.0Z
            </files:Instance_Creation_Date_Time>
13        <files:creator>
14          <files:Person_Name>
15            <about:PreferredGivenName>Robert</about:PreferredGivenName>
16            <about:MiddleName>C</about:MiddleName>
17            <about:FamilyName>Leif</about:FamilyName>
18            <about:qualification>Ph.D.</about:qualification>
19            <about:Generation/>
            </files:Person_Name>
          </files:creator>
        </list:File_Information>
20      <list:List_Mode_Componants>
21        <list:Num_Events>10000</list:Num_Events>
22        <list:Num_of_Channels>15</list:Num_of_Channels>
23        <list:Parameter_Info>
24          <channels:Parameter >FL1-A</channels:Parameter>
25          <channels:Long_Name>Fluorescein-AntiCD4</channels:Long_Name>
26          <channels:Channel_Number>3</channels:Channel_Number>
          </list:Parameter_Info>
        </list:List_Mode_Componants>
      </list:FCS>
```

Element 1 of Code Fragment 2 (Binary_Data_Info) starts the description of the contents of the binary data containing file. Element 2 starts the description of the Acquisition_Context. The first item in this description (Element 3) is the Acquisition number, which is a number produced by the instrument that is incriminated when the binary data is acquired. The Instance_Number (Element 4) starts at 1 and is Incremented for each binary file that is included in the ACSC file. Element 5 (Instance_Role) describes, as is shown in Fig. 3, the storage of the binary data containing files, the ToC and other meta-data. If only a single set of data is acquired, then all of the information is stored in the same ACSC file. In the case of multiple closely related data files where only a few simple variables have been changed, it is still possible to store all of the information in one ACSC file. These special cases are either a measurement that has one or more controls or a series of measurements where multiple data files are created and only one or a few simple variables have been changed. Examples of these simple variables are the time, voltage on a PMT or drug screening studies where the effects of different compounds and their concentrations have been changed or controls are measured. Example of variables where the data files should not be included in one ACSC file are: measurements on multiple patients or protracted time periods that involve multiple repeat measurements of the same standards to accommodate the possibility of the settings changing with time. When the Instance_Role is a variable (Element 6), the value (Element 7) and name (Element 8) of the variable for each measurement is included in the Acquisition_Context part of the Binary_Data_Info element of the ToC. Otherwise each data file and its ToC are put into their own ACSC. The meta-data that is constant for the series of measurements is included in a series file together with the URI of each member of the series. The meta-data that changes for each measurement is included in the ACSC. The proceeding stipulations are for circumstances where there is a significant hazard associated with an incorrect connection of meta-data to binary data. These stipulations possibly could be relaxed, if evidence were obtained that this hazard had been mitigated by another solution and/or for applications that were not of the criticality of patient care or the preparation of forensic evidence. For example, there should be no restrictions on how data files may be combined in ACS containers for research purposes other than those imposed by regulatory agencies.

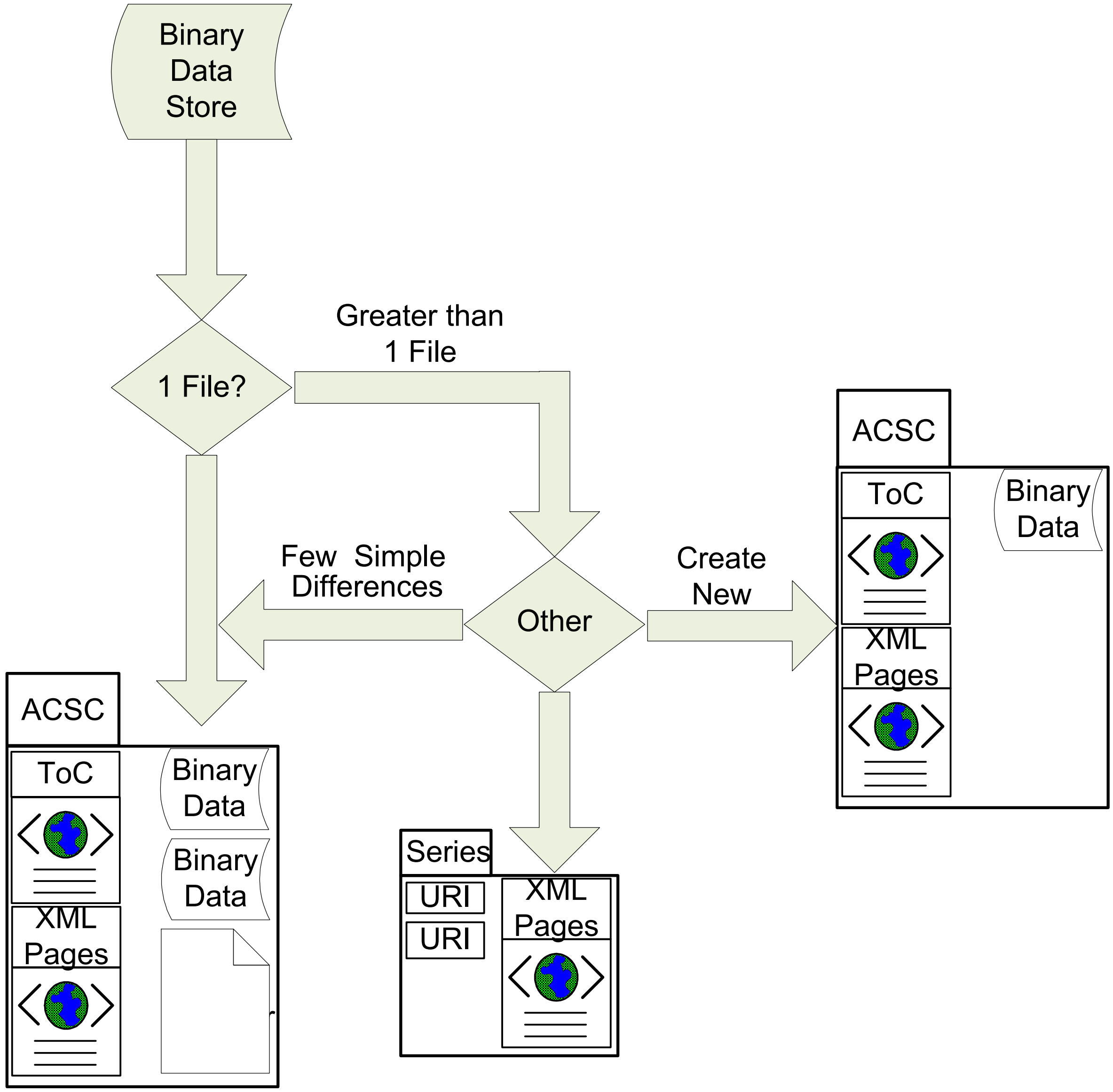

Fig. 3 is a flow chart that describes the process of determining the Instance_Role (Element 5). As shown in the flow chart, the Instance_Role determines if a binary data containing file can be stored in an ACS container (ASCS) or if it is also necessary to have information pertaining to the data file included in a series file. If a single data file has not been designated as part of a series, then it and its ToC are stored in an ACSC together with the other meta-data. If there is greater than 1 file and the only differences between the files are essentially a few attributes or elements that are each based upon a simpleType, such as being a control or a single variable, the data containing file together with its ToC file and other meta-data are also stored in the same ACSC as the original data file. If multiple data files have changes in more than a few single variables, a new ACSC file is created for each file, its URI is added to a Series file and the data file and its ToC are stored within the new ACSC. The other meta-data files are sent to the Series file only if they are the same for all of the data files; otherwise, the meta-data files that vary in their content are stored in the new ACSC.

The rest of Code Fragment 2 describes the format of the binary data containing file. In this case, it is a list-mode_File_Info (Element 9). The other two possibilities are an image file (Image_File_Info) and an other file (Other_File_Info). Each of these 3 possibilities (elements) is described by its own complexType. These complexTypes are respectively imported from the list_mode.xsd, image.xsd, and files.other.xsd schemas. Since these complexTypes have been hidden in other schemas, their descriptions do not clutter up ToC.xsd. The List_Mode_File_Info_Type includes a choice of the 3 types of list-mode containing files: FCS, NETcdf, and RAW. Because of its present popularity, FCS was selected for this example (Element 10). Since each of the various file types has a different file extension, the extension fcs is the value of the attribute File_Extension (Element 10). However, the 3 types of list-mode files include common elements. Please compare the compact notation employed for assignment of the value (fcs) of the attribute, File_Extension, in Element 10 with the wordy notation employed for assignment of a value (PMT Volts) of Element 8 (Variable_Name).

Each of these datatypes and all of the rest of the File_Info_Type(s) in the ToC schema includes the same type of File _Information (Element 11) which is imported from the files.xsd schema. This information includes attributes that contain answers to the following questions: Is this the data directly obtained from the measurements (original)? Is this information required to process the data? What is the status of the text in the file? What is the file's name? What is the URI (location) of the file? It also includes a place for a description of the file format. Lastly, binary data files include a choice between a pure binary (Raw) file and a mixed file, such as FCS, which includes both text and binary information. List_Mode_File_Info also includes Element 12, which provides the date and time the file was created.

Element 13 describes the creator of the file, which can be a human or a computer. In Code Fragment 2, it is a person (Element 14). The full name of the person in the form of separate elements 15,16,17,18, and 19 was provided in order to facilitate software matching of the person's name. At the end of Element 13, which occurs after Element 19, the File_Information element ends. This File_information, which is included in Elements 11 through 19 with some variations, principally the files' content, format, and extension together with other data-types from files.xsd has been reused in the other File_Info_Types.

Element 20 (list_Mode_Components) includes the number of events (particles or cells) recorded (Element 21) and the number of Channels or parameters (Element 22) contained in each event.

The Parameter_Info (Element 23) is a short description of each parameter (Channel in DICOM), which is independent of the list-mode file format. A parameter is a measurement performed by an instrument. Each type of parameter is identified. A Parameter (Element 24) describes what was detected, such as FL1-A, which is an abbreviation for fluorescence detector 1 pulse area. This is followed by a Long_Name (Element 25), which provides a longer, more readable description, such as, Fluorescein-AntiCD4. The last element (26) is Channnel_Number, which also ends the description of an FCS file.

Since a description of each type of parameter is required, this XML file should include all 15 (Element 22) parameter descriptions.

The element Index_File_Info (Not shown in Code Fragment 2) is the final part of Binary_Data_Info. Index_File_Info describes a binary file that is an array of positive integers. This description includes information from the files:File_Info_Type. The value of the File_Content attribute is set to Binary. An element Indexing_Parameter that identifies a cell or other particle type and the number of events (positive integers) in the file. The value of each integer is the position (index) in the list-mode file array of the record of Electronic_Events that has been found to describe a specific type of cell. Looping through an index is far more efficient than scanning each element of a list-mode file for a cell type[11].

3.1. Description of the Individual Elements in Code Fragment 1 from the ToC Schema

Returning to ToC_Type in the toc.xsd schema. The individual elements will be described below and to facilitate this description they are repeated above the paragraph that describes them.

```
4. <element name="Series_Info" type="files:XML_File_Type"/>
```

Series_Info points to a Series file, which includes information, similar to that in ASCS file except that it describes two or more binary data files. A series file contains information that is common to a group of measurements. Therefore elements in the ACSC should be able to be elevated to the series level when the Sample_Specimen(s) are essentially identical for the measurements and the same instrument is used. The one element that cannot be included in the series file is the Binary_Data. The series.xsd schema is based on A DICOM series in the context of cytometry.

The information in the Series_Info element includes: information from the files:File_Info_Type. The XML URI in the File_Info_Type is complemented by a DICOM Unique Identifier (UID), which is a string consisting of up to 64 numeric characters and the period character. The UID should assist software that permits a Series_Info element in the ToC to point to a value that is located in a DICOM file that contains a series. The Series_Info element also includes the series number, which is a 32 bit integer. The value of the File_Content attribute is set to either XML or Other.

```
5. <element name="Instrument_Data_Info"
      type="toc:Instrument_Data_Info_Type"/>
```

The Instrument_Data_Info element describes which type of instrument was used: flow cytometer, digital microscope, or other instrument. The actual instrument schema includes the settings of the instrument for the measurement(s). In the case of a flow cytometer, the instrument can be described in an XML file, which can be validated by the flow instrument schema, included in the binary data file (FCS), an other file, or is nonexistent. The Instrument_Data_Info includes: information from the files:File_Info_Type, a unique identifier (short textual description), the names of the manufacturer and model, and an optional description of the instrument.

```
6. <element name="Compensation_File_Info"
      type="toc:File_Format_Modifiable_Data_Type"/>
7. <element name="Gating_Method_Info"
      type="toc:File_Format_Modifiable_Data_Type"/>
8. <element name="Analysis_Method_Info"
      type="toc:File_Format_Modifiable_Data_Type"/>
```

The Compensation_File_Info, Gating_Method_Info, and Analysis_Method_Info elements are of the type toc:File_Format_Modifiable_Data_Type. The binary data can be changed by the operations described by the schemas that describe the compensation gating or analysis of the binary data, which are referenced by elements of this data-type. These elements each specify their own existence. If they exist, they describe whether the binary data has already been respectively compensated, gated and/or analyzed. Each of these elements also provides information from the files:File_Info_Type. The actual descriptions of the compensation, gating, and analysis can be located within a binary data containing file, such as FCS, be separate in an XML or other type of file.

```
9. <element name="Experiment_Overview_Description"
      type="toc:XML_or_Other_File_Type"/>
10. <element name="Sample_Specimen_Details"
      type="toc:XML_or_Other_File_Type"/>
11. <element name="Result_File_Description"
      type="toc:XML_or_Other_File_Type"/>
12. <element name="Report_file_Description"
      type="toc:XML_or_Other_File_Type"/>
```

Experiment_Overview_Description_Info, Sample_Specimen_Details_Info, Result_File_Info, Report_File_Info are all to point to either an XML_File_Type or to an Other_File_Type. These Info_Files all include information from the files:File_Info_Type. The result and report files could be in the form of an office type product. The latest versions of the most common applicable files are based on XML[22-29] and thus files using these format could be considered to be XML files.

```
13. <element name="Other_File_Info"type="toc:XML_or_Other_File_Type"/>
14. <element name="Additional_Info" type="anyType" minOccurs="0"
      maxOccurs="unbounded"/>
```

These two elements a place holder to permit extension of ToC.xsd

4. DISCUSSION

ISAC including the DSTF and other interested parties is composed of individuals with differing interests including: engineering, chemistry, algorithm development, software engineering, biomedical research, and clinical practice. This has led to a diversity of opinions on the requirements and content of the ACS. Determinations had to made of which meta-data needed to be captured; and how the software that contains this information should be organized. Obtaining consensus under these circumstance was and is a challenging issue and fortunately an educational experience. Consequently, the development of the toc.xsd and other schemas was and is an iterative process that involved considerable discussion amongst the authors and other members of the DSTF. In a significant number of cases, elements were moved from one schema to another. The only changes that had to be made were the additions to the schemas in the xmlns attribute list of visible schemas and to the list of imported schemas. The prefixes that abbreviated each added schema's URI had to be changed in schemas that used the data-types that had been moved. Locating these fixes was greatly facilitated by the failure of the schema parsers to validate until the essential corrections were made. Although both schema development tools (XMLSpy and Stylus Studio) would find the approximate location of the errors, neither would produce error messages equivalent to those produced by a good compiler. The incapacity of the present version of XSDL to restrict complexTypes in different schemas than their origin significantly interfered with the use of generics (templates) as a form of inheritance.

The ToC element of the schema provides a concise description of the entire ACS collection of schemas, which would not have been possible if this complexType had been interspersed with extraneous type definitions. Each specified element in the sequence contained in the ToC_Type is to be or has been instantiated by the creation of a corresponding schema. In each of these schemas, there is a similar complexType that describes the contents of the entire schema. Many of the data-types in the CytometryML schemas including toc.xsd have been reused from DICOM. The addition of attributes to reference DICOM Tags and Value Representations (data-type abbreviations) that reference specific reused DICOM data-types results in the use of elements that are based on complexTypes; rather than attributes, which can only be are based on simpleTypes. However the benefits of the reuse of data-types from another standard (DICOM) outweighs the syntactic inconvenience of using elements instead of attributes. It is anticipated that data-types from other standards will continue to be reused. The benefits of reuse include significant decreases in design and documentation costs, increased potential interoperability, and the possibility of influencing other groups to approach the transition of their standards to XML in a similar manner, which will facilitate interoperability. This interoperability will lessen the standards development burden on ISAC and similar societies because very few if any of the members of an individual organization have sufficient domain knowledge to construct an entire medical and scientific information standard[30]. The use of a modular design in the development of the CytometryML schemas should facilitate their replacement by other scientific societies who have greater domain knowledge than that of the present ISAC developers.

The schema for the Table of Contents construct of a ZIP file based container was reasonable to implement. Experience with implementation of a ToC for list-mode flow cytometry files indicates that image files together with their meta-data could also be stored in an ACSC file and perhaps stored together with the list mode data acquired from a series of images whose only difference was their location on the microscope slide. Comprehensive coverage of the available variety of image formats and compression techniques will be a significant challenge.

5. CONCLUSIONS

A reasonable solution has been proposed to the vexing question of when a measurement instance is part of a series, which requires the creation of a separate series file, and when the use of a single ACSC is sufficient. Guidelines have been proposed on when an element should be included in either the ACSC or the Series files.

Elements have been reused either directly or with slight modification. The problem of the multiplicity of file types that can serve the same purpose has been adequately handled by employing schema choice elements. The design and implementation of ToC.xsd is sufficiently mature that it can now be used to provide an overview of the rest of the software development process including the development of a schema that describes a microscope and its associated schemas.

The modularity of the CytometryML schemas and their existing content has facilitated the rapid creation of new schemas, such as ToC.xsd, instrument.flow.xsd, and series.xsd. It has been possible to develop code in XSDL that is very similar to the data-type definitions in the classes or packages of current higher-level programming languages.

The effective reuse of DICOM data-types, design, and documentation has significantly shortened the development effort and time. Especially, when used in conjunction with standard object-oriented design technology. Adherence to Req. 2, producing sufficient data to reproduce an experiment appears to be possible.

6. ACKNOLEDGEMENTS

This work was sponsored by Newport Instruments and National Institute of Biomedical Imaging and Bioengineering; Grant number: EB005034. RRB is a Research Scholar of Michael Smith Foundation for Health, and an ISAC Scholar. One of us, RCL, wishes to thank the members of DICOM Working Groups 26 and 27 for extending his knowledge of DICOM. However, any mistakes concerning DICOM are his own.

7. REFERENCES

1] Spidlen J., Leif R. C., Moore W., Brinkman R., “Analytical Cytometry Standard (ACS) Container Usage Recommendation," Available at: http://flowcyt.sourceforge.net/netcdf/

2] 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://superb-east.dl.sourceforge.net/sourceforge/flowcyt/Requirements-v070920.pdf (2007).

3] Lee J. A., Spidlen J., Boyce K., Cai J., Crosbie N., Dalphin M., Furlong J., Gasparetto M., Goldberg M., Goralczyk E. M., Hyun B., Jansen K., Kollmann T., Kong M., Leif R. C., McWeeney S., Moloshok T. D., Moore W., Nolan G., Nolan J., Nikolich-Zugich J., Parrish D., Purcell B., Qian Y., Selvaraj B., Smith C., Tchuvatkina O., Wertheimer A., Wilkinson P., Wilson C., Wood J., Zigon R., Scheuermann R. H., and Brinkman R. R., “MIFlowCyt: The minimum information about a flow cytometry experiment," Cytometryt A 73A, 926-930, (2008).

4] Leif R. C., “CytometryML and Other Data Formats," Proc. SPIE 6088-0L, 1-7 (2006).

5] Leif R. C., Spidlen J., Brinkman R. R., “Cytometry standards continuum," Proc. SPIE 6859, 68590Q-1-8 (2008).

6] Analytical Cytometry Standard (ACS) Container Usage Recommendation, Available at: http://flowcyt.sourceforge.net/acs/latest.pdf.

7] Berners-Lee T., Fielding R., and Masinter L., RFC 3986: “Uniform Resource Identifier (URI): Generic Syntax. January 2005”. Available at: http://www.ietf.org/rfc/rfc3986.txt

8] Seamer L. C., Bagwell C. B., Barden L., Redelman D., Salzman G. C., Wood J. C., Murphy R. F., “Proposed new data file standard for flow cytometry, version FCS 3.0”. Cytometry 28, 118–122 (1997).

9] Spidlen J., Leif R.C., Moore W., Brinkman R.R., “Analytical Cytometry Standard: NetCDF Conventions for List Mode Binary Data File Component (ISAC Candidate Recommendation)”. Available at: http://www.isac-net.org/media/standards/Conventions_080102.pdf.

10] Unidata, NetCDF (network Common Data Form). Available at: http://www.unidata.ucar.edu/software/netcdf/.

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

12] 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).

13] Parnas D. L., “On the Criteria To Be Used in Decomposing Systems into Modules," Communications of the ACM 15, 1053-1058 (1972).

14] Parnas D. L., Clements P. C., Weiss D. M., “Enhancing reusability with information hiding," ITT Proceeding of the Workshop on Reusability in Programming, cse.msu.edu. (1983).

15] Boehm B.W., “A spiral model of software development and enhancement," IEEE Computer, 61-72 (1988).

16] Digital Imaging and Communications in Medicine (DICOM) Part 3: Information Object Definitions, PS 3.3-2006, National Electrical Manufacturers Association, NEMA, Available at: http://medical.nema.org/dicom/2008/ (2008).

17] XML Schema Part 1: Structures Second Edition W3C Recommendation 28 October 2004. Available at: http://www.w3.org/TR/xmlschema-1/ (2004).

18] Leif R.C., Leif S.H., Leif S.B., “CytometryML, a markup language for analytical cytology," Proc. SPIE 4962, 288-297 (2003).

19] Pianykh O. S., [Digital Imaging and Communications in Medicine (DICOM): A Practical Introduction and Survival Guide], Springer Publishers, Berlin & Heidelberg, ISBN 354074570X, 9783540745709 (2008).

20] XML Schema Part 2: Datatypes Second Edition, W3C Recommendation 28 October 2004 Available at: http://www.w3.org/TR/2004/RECxmlschema-2-20041028/ (2004).

21 Warmsley P., [Definitive XML Schema], Prentice Hall, Available at: http://www.phptr.com (2002).

22] Standard ECMA-376, Office Open XML File Formats, December 2006, Available at: http://www.ecma-international.org/publications/standards/Ecma-376.htm (2006).

23] ISO/IEC 29500-1:2008 Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 1: Fundamentals and Markup Language Reference. Available at: http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html (2008).

24] ISO/IEC 29500-2:2008 Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 2: Open Packaging Conventions. Available at: http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html (2008).

25] ISO/IEC 29500-3:2008 Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 3: Markup Compatibility and Extensibility. Available at: http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html (2008).

26] ISO/IEC 29500-4:2008 Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 4: Transitional Migration Features. Available at: http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html (2008).

27] ISO/IEC 29500-1:2008, Information technology -- Document description and processing languages -- Office Open XML File Formats -- Part 1: Fundamentals and Markup Language Reference, Available at: http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=51463 (2008).

28] ISO/IEC 26300:2006, Information technology -- Open Document Format for Office Applications (OpenDocument) v1.0 (2006).

29] Open Document Format for Office Applications (OpenDocument) v1.1, OASIS Standard, 1 Feb 2007 Available at: http://docs.oasis-open.org/office/v1.1/OS/OpenDocument-v1.1.pdf (2007).

30] Leif R. C., “CytometryML: a data standard which has been designed to interface with other standards," Proc. SPIE 6441, 64410P 1-11 (2007).