CytometryML, a data standard, which has been designed to interface with other standards

Robert C. Leif*

XML_Med, a Division of Newport Instruments, 5648 Toyon Road, San Diego, CA, USA 92115-1022

ABSTRACT

Because of the differences in the requirements, needs, and past histories including existing standards of the creating organizations, a single encompassing cytology-pathology standard will not, in the near future, replace the multiple existing or under development standards. Except for DICOM and FCS, these standardization efforts are all based on XML. CytometryML is a collection of XML schemas, which are based on the Digital Imaging and Communications in Medicine (DICOM) and Flow Cytometry Standard (FCS) datatypes. The CytometryML schemas contain attributes that link them to the DICOM standard and FCS. Interoperability with DICOM has been facilitated by, wherever reasonable, limiting the difference between CytometryML and the previous standards to syntax. In order to permit the Resource Description Framework, RDF, to reference the CytometryML datatypes, id attributes have been added to many CytometryML elements. The Laboratory Digital Imaging Project (LDIP) Data Exchange Specification and the Flowcyt standards development effort employ RDF syntax. Documentation from DICOM has been reused in CytometryML. The unity of analytical cytology was demonstrated by deriving a microscope type and a flow cytometer type from a generic cytometry instrument type. The feasibility of incorporating the Flowcyt gating schemas into CytometryML has been demonstrated. CytometryML is being extended to include many of the new DICOM Working Group 26 datatypes, which describe patients, specimens, and analytes. In situations where multiple standards are being created, interoperability can be facilitated by employing datatypes based on a common set of semantics and building in links to standards that employ different syntax.

Keywords: CytometryML, DICOM, FCS, flow cytometry, pathology, informatics, standard, XML, schema, RDF

1. INTRODUCTION

The process of creation of cytology-pathology standards is accelerating. These standards must interoperate and their use must not be limited to single societies or groups. Standards should be inclusive rather than exclusive. Both clinical and research cytometry data must be supported and the extra requirement of clinical cytometry of exchanging data with hospital information systems must be met. The list-mode and image data from flow cytometry, digital microscopy including pathology images, new analysis techniques, and much of the related data that describes it (metadata) must be integrated with the patient's clinical information.

This integration of cytometry data with clinical information systems would be greatly facilitated by the use of a common data standard or collection of interoperating standards by the interested societies or groups. However, since the College of American Pathologists plans to use the Digital Imaging and Communications in Medicine (DICOM) standard and the International Society for Analytical Cytology (ISAC) does not want to use DICOM, interoperability can still be achieved if all the groups use the same set of datatypes and any new standard is based on the eXtensible Markup Language, XML.

The labor involved in creation of a cytometry standard can be significantly decreased by employing the same standard for flow and image cytometry. It is reasonable to apply a common data standard to these two modalities, since the differences between the software models of a digital microscope and a flow cytometer are minimal and both modalities are employed in the same laboratory for similar purposes.

Figure 1 shows the proposed relationships between the DICOM and XML parts of the laboratory-hospital information system. The data store could be part of either the laboratory or the hospital information system.

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

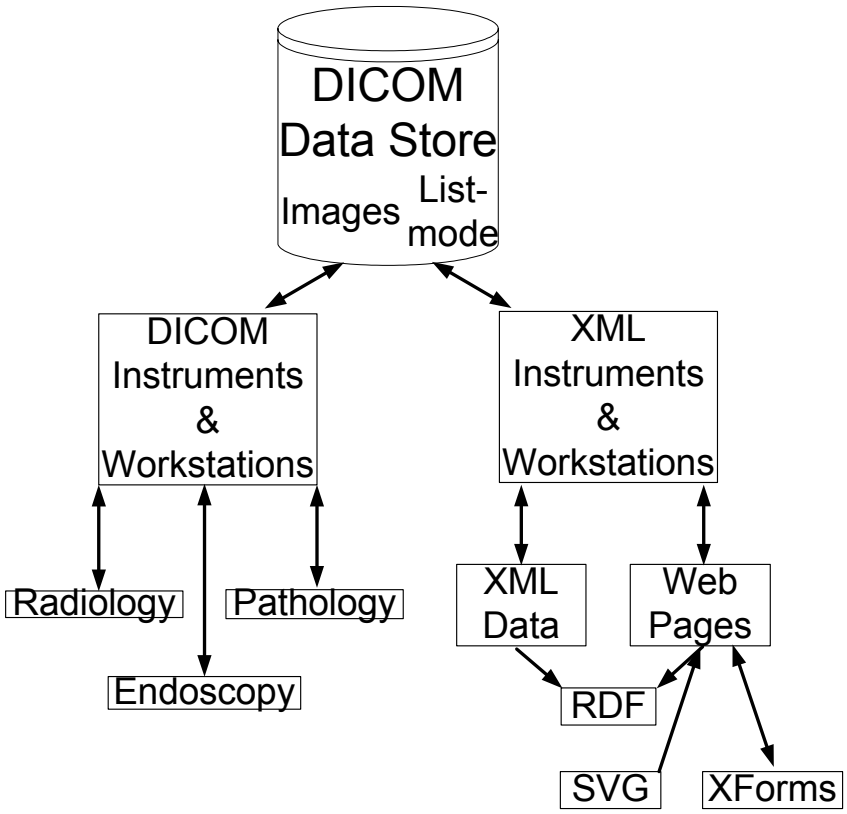

Figure 1, Diagram showing the interconnections between the DICOM and XML. The DICOM data store collects the image and list-mode data from both the DICOM and XML instruments.

As shown in the Figure 1, the DICOM based instruments including digital microscopes will, as at present, create and transmit their results to the data store, which will then be able to exchange this data with the laboratory and hospital information systems. Flow cytometry XML and list-mode binary data could also be transferred and retrieved from this store. This interconnection design provides the following benefits: 1) The combination of image and list-mode data obtained with either a digital microscope or an imaging flow cytometer will be kept together. 2) Laboratories that use both modalities will be able to store their data in the same place. 3) The storage of the data in a common server system will satisfy the needs of many of the pathologists and physicians from other specialities and will be consistent with the interests of the hospital informatics department. 4) The part of the data that is created by a human can be entered using a XML standard form description language, XForms (1), which has provisions to minimize data entry errors.

The analyzed data will be presented either using XML in the form of an office suite or as an XHTML based web page. Scalable vector graphics, SVG (2) is a portable web standard, which can be used to present cytometry data in the form of graphs.

The relationships between the data will be codified with the Resource Description Framework, RDF (3). RDF is a language for representing information about resources, particularly metadata. For instance for Web documents, this could be: title, author, and modification date, copyright and licensing information. RDF is intended for the processing of information by applications, rather display. It uses Uniform Resource Identifiers, or URIs) to identify objects. It also requires an ontology, which is an agreed upon, controlled formal vocabulary and grammar for knowledge domains that describe objects and the relations between them. A flexible tool like RDF is essential for research, which by its nature must be free to change the characteristics of a test. Since clinical data must be obtained using predefined tests, the use of RDF probably will be unnecessary for the creation of the laboratory data. However, the use of RDF attributes and elements can be very helpful when studying the results of clinical and research studies and organizing reports. Examples of RDF applications (4,5) can be found in the Ontology for Biomedical Investigations (OBI, formerly FuGO) project, which describes itself as

"developing an integrated ontology for the description of biological and medical experiments and investigations. This includes a set of 'universal' terms, that are applicable across various biological and technological domains, and domain-specific terms relevant only to a given domain." Its stated purposes are: "support the consistent annotation of biomedical investigations, regardless of the particular field of study."

Three complementary development projects should be able to provide this proposed integration of XML and DICOM. These projects are the extension to Digital Imaging and Communications in Medicine, DICOM (http://medical.nema.org/dicom), being developed by Working Group 26, Flowcyt (http://flowcyt.sourceforge.net/), and CytometryML (http://www.newportinstruments.com/cytometryml/cytometryml.htm). Working Group 26 is proposing a pathology extension to DICOM, Supplement 122. Flowcyt is developing: a simpler and consequently safer and easier to parse version (6) of the Flow Cytometry Standard (FCS) and schemas for gating as well as compensating and transforming flow cytometry data. CytometryML is a set of XML schemas that will incorporate the datatypes that will be present in Supplement 122 that are relevant to analytical cytology, describe other datatypes that are specific for analytical cytology, and has already been able to import some of the Flowcyt gating schemas and should be able to import the rest. The purpose of CytometryML schemas was and is to precisely specify datatypes that can be used to facilitate: the transfer, storage, presentation, and creation of data, while minimizing the probability of mistakes that can occur during these processes. These uses include: describing in detail objects, such as: images, parts of a microscope, flow cytometer, slides, staining, and binary data (7,8) that describes individual cells; assisting in the design and creation of databases; and providing data in a form suitable for reports and forms.

The three complimentary approaches differ in their emphasis. Flowcyt is directed towards flow cytometry, particularly gating and analyses. DICOM Supplement 122 towards microscopy, principally of tissues with emphasis on whole slide imaging. CytometryML is an attempt to create a standard for cytometry that is equally applicable to flow and image cytometry, and also can be used for both clinical and research data. CytometryML has provided and will provide a subset of DICOM in XML syntax that can be combined with Flowcyt and RDF ontologies such as FaceOntology (http://flowcyt.sourceforge.net/ontology/) and the Laboratory Digital Imaging Project, LDIP, (http://www.ldip.org). The FaceOntology is to be incorporated into the Ontology for Biomedical Investigations, OBI (R. R. Brinkman, personal communication).

Previously, the capacity of data standards to interoperate has been confounded by the differences in the requirements, needs, and past histories of the creating organizations. One of the reasons for the creation of CytometryML was to provide a practical approach to achieve interoperability. This approach is based on the concept of reuse, which is a well known software engineering practice. Reuse besides being applied to code, has been applied to many other parts of the development environment, such as designs, documentation, and tests (9). Standards paucity, the practice of reusing datatypes and their documentation from other standards, is an extension of reuse methodology to standards. This reuse minimizes the design effort, facilitates interoperability, and maximizes the reliability of the CytometryML schemas due to the previous successful use of many of the datatypes in implementations of DICOM and FCS.

Although the different groups employ different representations (syntax) for their data, the definitions (semantics) of the datatypes should, as much as possible, be common to all of the standards. The achievement of interoperability should be facilitated by employing datatypes with common semantics and limiting the difference between standards to syntax. The extension of DICOM by Working Group 26 will result in the significant benefit of having one set of semantics and terms for medical imaging that is used throughout the medical profession and by scientists engaged in cytometry. CytometryML includes an attempt to create a pilot implementation in the XML Schema Definition Language, XSDL, of part of the designs developed by Working Group 26.

2. METHODS

A requirements document (10) and a hazard analysis (11) were published to acquire appropriate peer review. Datatypes were reused from DICOM and FCS (12), and numerical types were reused from Ecma-International (http://www.ecma-international.org/). The CytometryML schemas were developed using the XML schema language, XSDL (13,14), and were validated with both StylusStudio (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 (15) and datatypes that could be part of an extension of DICOM datatypes.

3. UNITY OF ANALYTICAL CYTOLOGY

A generic instrument schema, which contained an Instrument_Type was created, which is based on datatypes imported from multiple schemas. Both a Microscope_Type and an Flow_Cytometer_Type were created by restriction from the generic Instrument_Type (Figure 2). XMLSpy was used to produce example XML files (documents) from the Microscope and Flow_Cytometer elements. Use cases were created by filling these XML pages with values and validated with pages that were created with XMLSpy. Visual inspection of the XML document was used to detect design defects including the order of the elements. The appropriate schema was corrected and the process was iterated. This process was first described by Boehm; and he named it, the spiral model of development (16)

.

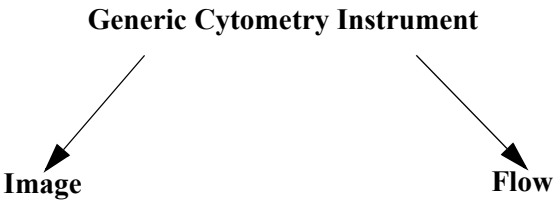

Figure 2. Derivation of image and flow cytometers from a generic cytometry instrument

3.1. Description of the CytometryML Instrument schema datatypes

The schema example below is color coded in the electronic version. XSDL is a nested language that begins statements which the less than character **<** to begin a construct (element) and ends constructs that are not nested with **/>**. Schemas include both simpleTypes and ComplexTypes. SimpleTypes are data structures that can only be based on a single elementary type like a string or a number. Attributes are a shorthand description of an object that can be only based on a simpleType. An element is a general purpose description of all other types of objects. ComplexTypes are data structures that include multiple entities or one or more attributes.

The first step is to create simple (helper) datatypes

```
S1.<simpleType  name="Platform_Type" id="Platform_Type">
S2.     <restriction  base="token">
S3.         <enumeration  value="Upright"/>
S4.         <enumeration  value="Inverted"/>
S5.         <enumeration  value="Plainer"/>
S6.     </restriction>
S7.</simpleType>
```

Statement numbers have been added at the far left and will be referred to in parenthesis. Statement (**S1**) employs the less than character **<** to begin a construct. The first element is a user defined **simpleType**. Since (**S1**) begins a set of nested statements it ends with the **>**. The word **name** is an **attribute** that has a value **"Platform_Type"**. Attributes are always based upon simple_Types. The value of an attribute is set in quotation marks and follows the equals sign. As described in Section 5.1. Approach 5 below, the id attribute has been included in the hope of making this datatype accessible to RDF.

The **Platform_Type** is derived by restricting (**S2**) a previously defined type of string a **token** (a string that does not include leading or trailing spaces and several nonprinting characters).

As shown on (**S3**), statements that are not nested end with **/>**. Statements (**S3**, **S4**, and **S5**) provide the values for this enumerated type. In XML, enumerated types are restrictions of the type string. The list of enumeration values ends at (**S5**).

The **restriction**, which is a nested statement, ends (**S6**) with a **</** that is followed by repeating the first word of (**S2**) and is ended by the **>** character. The ending of the **simpleType** (**S7**) follows the same format.

The **Viewing_Type** employs similar syntax

```
S1<simpleType name="Viewing_Type">
S2.    <restriction base="token">
S3.        <enumeration value="Mono"/>
S4.        <enumeration value="Sterio"/>
S5.        <enumeration value="Other"/>
S6.    </restriction>
S7.</simpleType>
```

A complex datatype (**complexType**) is created to model a generic instrument

```
S1.<complexType name="Instrument_Type">
S2.  <sequence>
S3.    <element name="Item_General_Info" type="item:Item_General_Info_Type"/>
S4.    <element name="Objective" type="optics:Optic_Type" minOccurs="1"
         maxOccurs="7"/>
S5.    <element name="Condenser" type="optics:Optic_Type" minOccurs="0"
         maxOccurs="10"/>
S6.    <element name="Platform" type="instr:Platform_Type"/>
S7.    <choice minOccurs="1" maxOccurs="1">
S8.        <element name="Stage" type="stage:Stage_Type"/>
S9.        <element name="Fluidics" type="fluid:Fluidics_Type"/>
S10     </choice>
S11.   <element name="Viewing" type="instr:Viewing_Type"/>
S12.   <element name="Sorts" type="boolean"/>
S13    <element name="Comments" type="dicom:Bd_1024_Type"
         minOccurs="0"/>
S14.  </sequence>
S15.   <attribute ref="optics:Objective_List"/>
S16.</complexType>
```

The first element (**S1**) is a complexType, because it includes a sequence (**S2**). A sequence contains one or more elements (**S3**-**S6**, **S11**-**S13**) each with its own type declaration. These elements can also be complexTypes. The type declarations in (**S3**-**S6**) include prefixes, such as **item:** and **optics:**. These prefixes point to the location of the schemas which contained them. Sequences can also include choice (**S7**) elements, which in turn can include other elements of which one can be selected to be included in the XML page. In this case the two elements are Stage (**S8**) and Fluidics (**S9**), which are complexTypes that have been created each with its own schema. Since the primitive type boolean (**S12**) is defined in the schema standard, it does not require a prefix. Elements **S4** and **S5** specify the cardinality, **minOccurs** and **maxOccurs,** of their elements. Attributes are based on simpleTypes and besides standing alone are used to describe and specify types. The terms **name**, **type**, **maxOccurs**, and **minOccurs** are all attributes. Attribute **S15** is a reference that describes a list that effectively contains the names (ids) of one or more microscope objectives.

Below, the generic instrument is transformed into a microscope and a flow cytometer. The numbers preceded by an M are statements describing a microscope, those preceded by an F are a flow cytometer, and those preceded by an S are common to both types of instruments.

```
M1. <complexType name="Microscope_Type">
F1. <complexType name="Flow_Cytometer_Type">
```

Statements **M1** and **F1** each start a complexType definition.

```
S2.    <complexContent>
S3.        <restriction base="instr:Instrument_Type">
S4.            <sequence>
S5.                <element name="Item_General_Info"
```

```
                type="item:Item_General_Info_Type"/>
M6.          <element name="Objective" type="optics:Optic_Type" maxOccurs="7"/
F6.          <element name="Objective" type="optics:Optic_Type" maxOccurs="1"/>
M7.          <element name="Condenser" type="optics:Optic_Type" minOccurs="0"
               maxOccurs="1"/>
F7.          <element name="Condenser" type="optics:Optic_Type" minOccurs="0"
               maxOccurs="10"/>
S8.           <element name="Platform" type="instr:Platform_Type"/>
M9.          <element name="Stage" type="stage:Stage_Type" minOccurs="1"
               maxOccurs="1"/>
F9.           <element name="Fluidics" type="fluid:Fluidics_Type"
              minOccurs="1" maxOccurs="1"/>
S10.          <element name="Viewing" type="instr:Viewing_Type"/>
S11.          <element name="Sorts" type="boolean"/>
S12.         <element name="Comments" type="dicom:Bd_1024_Type"
S13.           minOccurs="0"/>
S14.       </sequence>
M15.           <attribute ref="optics:Objective_List"/>
S16.     </restriction>
S17.  </complexContent>
S18. </complexType>
```

The differences between the two complexTypes are: The names in the first elements, **M1** and **F1**, of the complexType differ; elements **M6** and **F6** a microscope could have up to (**maxOccurs**) 7 objectives and a flow cytometer can have only 1; elements **M7** and **F7** a fluorescent microscope can have (**minOccurs**) 0 or 1 condensers and a flow cytometer can require a condenser for each light source (crossed cylindrical lenses); elements **M9** and **F9**, which are the two values of the choice element of the Instrument_Type; and the attribute (**M15**), which is only relevant to the Microscope_Type.

The the two values of the choice element of the Instrument_Type are each complexTypes. The microscope Stage complexType (**M9**) includes: The height (thickness) of the stage; the dimensions of the clear opening, the maximum velocity of the stage; the type of motor (servo or stepping); the resolution of the encoder; the motor step size; the accuracy of the stage position measurements; the repeatability of the stage position; the straightness and flatness of its travel; the maximum excursion in the X,Y, Z course and Z fine directions, the stage rotation; general information that describes the manufacturer etc.; and a description, if necessary. The units and ranges of all of the measurements are specified.

The flow cytometer Fluidics_Type (**F9**) includes: the velocity of the particles at the measurement point, the sheath and sample flow rates, and general information that describes the manufacturer etc., and a description, if necessary. The units and ranges of all of the measurements are specified.

4. XSDL SCHEMAS VS. DICOM

Since CytometryML is based on the principle of standards paucity, DICOM is the major source of the present datatypes and the datatypes created by DICOM Working Group 26 have and will be incorporated into CytometryML. These types include: the work order, specimen, tissue, collection procedure, container, slide, and coverslip. CytometryML schemas to describe preliminary versions of Working Group 26 design have already been created. These XSDL schemas include: the person, organ, specimen, stain, container, slide, and coverslip. Since DICOM was created prior to XML, DICOM has been extended (Part 18: Web Access to DICOM Persistent Objects (WADO)). However, DICOM does not appear to be able to access XML based objects. As shown previously (12), CytometryML includes attributes that link a XML schema complexType (class) to a DICOM datatype. This should facilitate the bidirectional transfer of data between DICOM and XML. If this bidirectional transfer can be achieved, then DICOM structured reporting (17) could be in the form of XML or XHTML documents, which are organized by one or more RDF pages. Data entry could be in the form of an XForm (1);

and DICOM could be augmented by the XML datatypes in CytometryML and the Flowcyt collection of schemas: Gating-ML (18), Transformation-ML (19), and Compensation-ML (20).

5. XSDL SCHEMAS VS. RESOURCE DESCRIPTION FRAMEWORK (RDF)

The uses of XML and HTML pages based on XSDL schemas is different from the uses of RDF. The present use of the RDF Vocabulary Description Language (RDFVDL) (21) to validate RDF elements is an issue that should be kept separate from the usage of RDF.

Since the Flowcyt Ontology for Flow Cytometry (http://www.flowcyt.org/FaceOntology/) and LDIP are being implemented in RDF and CytometryML schemas are being implemented in XSDL, it is appropriate to comment on the uses of the two technologies and the relationship between them.

The following is a list of some uses for XML metadata including that from cytometry measurements.

1. To be stored in either a standard SQL relational database and/or an XML database or a combination thereof;

2. to be transmitted from one computer to another;

3. to permit validation of data, particularly that entered by humans;

4. to publish the data as reports or papers;

5. to define control files, such as test definitions;

6. to be used to find relationships between data.

Schemas written in XSDL are useful for items 1-5. The elements and attributes needed to define the relationships (item 6) could also be described in XSDL schemas. The numbering of the following statements refers to the items in the uses described above.

1) The XMLSpy MapForce module can be used to develop relational databases based on XML schema. Both XMLSpy and StylusStudio have facilities to work with relational databases and convert them into XML.

2) Some transmission errors can be detected by validating the data with the same schema before and after transmission. These schemas could probably also be used for encoding and compression.

3) Human data entry, particularly dictation, is error prone. Schemas that include ranges and enumerations (strong typing) will catch entry errors. Ontologies are an obvious source for the datatypes and their definitions. Ontologies can provide the values for the enumerations. The limitation of the possible choices by employing strong typing cuts down on the number of possibilities that need to be considered by speech to text translators. This should significantly increase the accuracy of dictated data entry. The W3C standard for forms, XForms, works with XSDL schema.

4) Microsoft's Word 2007 and Excel 2007 are based on XML schema, which are part of the Open XML Format (http://msdn2.microsoft.com/en-us/library/ms406049.aspx); and other office suites will produce the Open XML Formats. The latest (26 July 2006)) W3C Working Draft of XHTML™ 2.0 (http://www.w3.org/TR/2006/WD-xhtml2-20060726/) is planned to accept XML Schema and should permit the use of XSDL for this purpose. The formatting of documents and forms could then be handled by either cascading style sheets, CSS (http://www.w3.org/TR/REC-CSS2/), or Extensible Stylesheet Language (XSL) Formatting Objects (XSL:fo, http://www.w3.org/TR/2006/REC-xsl11-20061205/).

5) Control files and test definition files require details that are of little to no utility for creating or inferring RDF relationships. However, the degree of detail in these files is necessary for reproducible operation and description of diagnostic and research systems. It is also one method of describing the datatypes in a software design.

6) If the determination of these relationships is to be accomplished outside of querying a database, then the functionality of RDF is definitely useful. Since RDF may be unfamiliar to the reader, an example of RDF and OWL from the FACE ontology is given below. OWL is the Web Ontology Language, which is an extension of RDF. Both employ standard XML syntax.

```
S1. <rdfs:Class rdf:ID="Flow_Cytometer_Descriptor">
S2.     <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
          Contains all information that describes parts of a flow cytometer. </rdfs:comment>
S3.     <rdfs:subClassOf>
```

S4 . <owl:Class rdf:ID="Instrumentation_Descriptor"/>
S5 . </rdfs:subClassOf>
S6 . </rdfs:Class>

The RDF-OWL example above is color coded in the electronic version. XML is a nested language that begins statements, as does XSDL, with the less than character < (S1), which is followed by a prefixed element name. The prefix **rdfs:** is an abbreviation that describes the source RDF schema that contains the RDF element **Class**. The **Class** element includes an attribute **rdf:ID** that identifies the class. Statement S2 is an element and thus is not an XML comment. This **rdfs:comment** employs an attribute, **rdf:datatype** to show the relationship of the attribute's value, which is a URI followed by the # character and the name of the datatype, which is a **string**. Thus the datatype of the **Flow_Cytometer_Descriptor** is a string. This is followed by the value of the element, which provides the information that is normally found in a comment. S2 is ended, as are all XML statements by ending the element with </ followed by repeating the element name, **rdfs:comment**, and ending with the > character.

S3 is an element that describes a relationship, **subClassOf**, of the **Flow_Cytometer_Descriptor** (S1). This **subClassOf** relationship has nested within it (S4) the identity of the parent class, **Instrumentation_Descriptor**. S5 ends the **subClassOf** element. S6 ends the **Class** element. The use of the **ID** attribute to identify objects precludes the direct use of XSDL complexTypes, since attributes can only describe simple types.

However, the syntax of RDFVDL schema (3,21) and the use of specialized tools could in many cases be replaced by tools that are based on XSDL schema (see below). An excellent use for RDF is to glue together and show the relationships between XML pages that have been validated against XSDL schema. This has been elegantly described by Brinkman and Spidlen et al. (22).

The essential difference between RDFVDL and XSDL schemas based applications is that RDF is used for showing connections and making inferences (artificial intelligence) and XML based on XSDL is used for standard information technology purposes. These uses have very different requirements. The means to relate a subject of a statement with an object in RDF do not require that either or both be strongly typed. In fact, the typing should probably be ignored by using something like the XSDL anyType construct. Whereas in conventional information technology applications, strong typing protects the integrity of data transmission, databases, data entry, data presentation, and data manipulation. In short, XSDL schemas are well suited to describe nouns and their associated adjectives and RDF is well suited to describe verbs. Since simple sentences include both, the ability of the two technologies to work together should be maximized. Because in science and medicine, the subject and object of a simple RDF statement can be complex entities, it should be possible for these to be validated against a XSDL schema.

The development and validation of XML based documents would be simplified if a common set of tools were employed. For instance, if XML based on both XSDL schema and RDFVDL schema were to exist on the same XML or XHTML page, the two parsers would have to work together. The lack a common schema language for both XSDL and RDFVDL schema is the source of this problem. The similarity in capabilities between the two schema languages indicates that it would have been possible and may still be possible to define RDF with XSDL based schemas (see Approach 3 below). One possible reason for the creation of another schema language is not invented here. In fact, theoretically an RDFVDL schema parser (23) should accept most of the simpleTypes in the XML Schema Standard; and a parser for the Web Ontology Language, OWL, could accept most user defined XSDL simpleTypes (24). Since many of the CytometryML ComplexTypes have corresponding SimpleTypes, they could be referred to by OWL. Unfortunately, RDF parsers are not required to accept XSDL complexTypes. Although two collections of schemas one in XSDL and the other in RDFVDL with a common set of datatypes is conceivable, the cost of duplicating the CytometryML XSDL in RDFVDL would be high and the verifying that the two sets of schemas were essentially identical would also be high. Besides basing the RDFVDL schemas on XSDL simpleTypes, there are several proposed solutions, which will be described below.

5.1. Possible Approaches to Achieve XSDL RDF Interoperability

1. The first approach is a conceptually simple one that Jules Berman (personal communication) is working on for the LDIP project (http://www.ldip.org). He is writing his own parser, which will work with XSDL complexTypes. In principle, much of the work in extracting the information present in the higher-level complexTypes in the CytometryML schemas could then be, at least, semi-automated.

2. The second approach is to apply a common model. The Flowcyt Fluorescent Activated Cell Experiment Ontology, FaceOntology (25), which is an ontology for flow cytometry (http://flowcyt.sourceforge.net/ontology/), includes many of

the CytometryML higher-level types. The FaceOntology is an RDF project written in OWL. However, it does not directly include in these higher level types the very detailed data in the lower-level elements and attributes that constitute these types. It does, however, include some of this detailed data as separate elements (22). This detailed data is required to totally reproduce settings on an instrument and to define laboratory chemistries. It is also necessary for clinical and pharmaceutical laboratories' documentation.

One major source of detailed data in CytometryML is the necessity to facilitate traceability and convertibility with preexisting standards, such as FCS 3.0 (26,27) and DICOM. Since the FaceOntology is an RDF application, it includes a linear inheritance model. Most of the major datatypes in CytometryML were created by composition. The element type pairs in the sequences came from multiple schemas. The example given above of the Microscope and Flow Cytometer being derived from a common ancestor (instrument) is atypical. The FaceOntology and the LDIP ontology both include excellent RDF comments, which will eventually be added to the corresponding CytometryML datatypes that are not based on DICOM datatypes.

3. A third approach that is relevant to items 3,4 & 6 is the use of RDFa. This has been discussed by Mark Birbeck (28) and is described in RDFa Primer 1.0 Embedding RDF in XHTML, W3C Working Draft 16 May 2006 (http://www.w3.org/TR/2006/WD-xhtml-rdfa-primer-20060516/). RDFa encodes RDF in web pages as patterns of HTML usage. Essentially a set of rules is provided to encode any RDF by introducing a set of RDF triples that are described by XML elements and attributes that are added to the text. These elements span relevant text strings, such as a person's name or telephone number. These XML elements contain an attribute which describes the relationship and has a qualified (prefixed) name, which points to its source. The text element is the value of the element.

```
<span property="foaf:name">Robert C. Leif</span>
```

FoaF is the prefix for Friends of a Friend Ontology (http://xmlns.com/foaf/0.1/).

The XHTML 2.0 Specification includes schema based modules that should include XHTML metainformation and XHTML metainformation attributes. This approach appears to work with XSDL simpleTypes and perhaps could be extended to complexTypes. Any text that is part of the document can be used as a value for an RDFa element; however presently, examples where the text was contained in an XML element have not been observed.

4. A fourth approach is based on the use of the Web Services Description Language (WSDL) (http://www.w3.org/TR/wsdl20-primer/). The primer states, "WSDL 2.0 processors are likely to support XML Schema at a minimum. However, WSDL 2.0 does not prohibit the use of some other schema definition language." The WSDL binding to RDF is described in the W3C Working Draft, Web Services Description Language (WSDL) Version 2.0: RDF Mapping (http://www.w3.org/TR/wsdl20-rdf/). The Web Service Semantics - WSDL-S W3C Member Submission (http://www.w3.org/Submission/WSDL-S/) demonstrates that it is possible to represent the semantics of a web service file by using XSDL complexTypes. The authors state, "The semantic information specified in this document includes definitions of the precondition, input, output and effects of Web service operations. This approach offers multiple advantages over OWL-S [http://www.w3.org/Submission/OWL-S/]. First, users can describe, in an upwardly compatible way, both the semantics and operation level details in WSDL- a language that the developer community is familiar with. Second, by externalizing the semantic domain models, we take an agnostic approach to ontology representation languages. This allows Web service developers to annotate their Web services with their choice of ontology language (such as UML or OWL) unlike in OWL-S. This is significant because the ability to reuse existing domain models expressed in modeling languages like UML can greatly alleviate the need to separately model semantics. Finally, it is relatively easy to update the existing tooling around the WSDL specification to accommodate our incremental approach."

Unfortunately, the authors' relatively easy procedure still appears to require considerable effort for complexTypes, Web Service Semantics - WSDL-S Technical Note, Version 1.0 April, 2005 (http://lsdis.cs.uga.edu/library/download/WSDL-S-V1.pdf) Section: Annotating Complex Types.

This approach has the advantage, that WSDL is an obvious way to transmit data from one computer to another. WSDL could provide a significant part of the functionality of DICOM and of equal significance could serve as a useful extension or means to interface other systems with DICOM.

5. A fifth approach described how to integrate OWL DL with user defined simpleTypes including the use of the id attribute (29). The simpleType can be referred to as the concatenation of the schema's URI reference, the # character and the value of the id attribute. For convenience, the value of the id attribute can be the same as the simpleType's name attribute. For instance the Platform_Type simpleType declaration above could be http://CytometryML/Schemas/instru-

ment#Platform_Type.

CONCLUSIONS

The essential unity of flow cytometers and digital microscopes has been demonstrated by deriving both datatypes from a common cytometer datatype. Since Flow cytometers and digital microscopes are sufficiently similar that they can be derived from a common ancestor, a generic cytometer; there is no need to have a separate standard for each modality.

In the development of standards, it has been possible to reuse designs and datatypes from other languages and environments. The creation of a collection of XML schemas, CytometryML, has demonstrated the feasibility of reusing the semantics of datatypes from DICOM and those from ISAC's FCS standard, as well as reusing their documentation. Similarly, in the case of RDF, the Flowcyt FaceOntology's reuse of many of CytometryML's datatypes has demonstrated the reuse of XSDL. XSDL has been used to rapidly prototype a WG 26 DICOM design.

It was previously demonstrated (12) that XSDL datatypes could include constant attributes to link CytometryML to the DICOM standard and the legacy Flow Cytometry Standard (FCS). However this inclusion results in a small but acceptable increase in complexity of the code, the use of XSDL complexTypes. Several methods have been described for combining XSDL with RDF. The problem of linking an XSDL complexType to RTF would disappear if RDF were validated against XSDL based schema instead of using RDFVDL schema. If the problem of the incompatibility of RDF and XML schema can be solved, the combination would be worth much more than the sum of the individual parts.

ACKNOWLEDGMENTS

I wish to thank Ryan Brinkman for his very helpful comments, suggestions and corrections to this paper. I also wish to thank the members of DICOM Standards Committee, Working Group 26 (Pathology) for sharing their designs, which I have tried to implement, and the following individuals for providing information on and explanations of their standardization efforts: Bruce Beckwith, Jules Berman, and Ryan Brinkman.

FINANCIAL DISCLOSURE

Newport instruments is owned by Robert C. Leif, Ph.D and his partners. Our plan is to offer royalty free licenses to scientific and medical societies provided that they either both charge a reasonable amount to sublicense the schemas and take over the responsibility for their maintenance; or the society sublicense the schemas for free with the proviso that the sublicense did not include a prohibition on software patents or other similar commercial limitations.

Copies of the XML files described above and the schemas used to generate it are available at http://www.newportinstruments.com/cytometryml/cytometryml.htm

REFERENCES

1. "XForms 1.0 (Second Edition)" W3C Recommendation 14 March 2006 (http://www.w3.org/TR/2006/REC-xforms-20060314/).
2. "Scalable Vector Graphics (SVG) 1.1 Specification, W3C Recommendation 14 January 2003" (http://www.w3.org/TR/SVG11/).
3. F. Manola, E. Miller (Editors), "RDF Primer", W3C Recommendation 10 February 2004, http://www.w3.org/TR/rdf-primer/
4. P. L. Whetzel, R. R. Brinkman, H. C. Causton, L. Fan, D. Field, J. Fostel, G. Fragoso, T. Gray, M. Heiskanen, T. Hernandez-Boussard, N. Morrison, H. Parkinson, P. Rocca-Serra, S. A Sansone, D. Schober, B. Smith, R. Stevens, C. J. Stoeckert, Jr., C. Taylor, J. White, A. Wood, and the FuGO Working Group, "Development of FuGO: An Ontology for Functional Genomics Investigations", *OMICS A Journal of Integrative Biology,* Vol. 10, pp 199-204 (2006).
5. "Ontology for Biomedical Investigations", http://obi.sourceforge.net/ (2007).
6. "A Proposal for FCS4", http://flowcyt.sourceforge.net/fcs/ (2006).
7. R. C. Leif, "CytometryML and Other Data Formats", in *Manipulation and Analysis of Biomolecules, Cells, and Tissues III*, D. Farkas, D. V. Nicolau, and R. C. Leif, Editors, SPIE Proceeding Vol. 6088-0L pp. 1-7 (2006).
8. R.C. Leif, "CytometryML, Binary Data Standards", in *Manipulation and Analysis of Biomolecules, Cells, and Tissues II*, D. Farkas, D. V. Nicolau, and R. C. Leif, Editors, SPIE Proc. Vol. 5699, pp. 325-333 (2005).
9. B. Boehm, K. Sullivan, "Software Economics: A Roadmap", in *International Conference on Software Engineering, Proceedings of the Conference on The Future of Software Engineering*, ACM Press New York, NY, USA. (2000).
10. R. C. Leif, S. B. Leif, and S. H. Leif, "CytometryML, An XML Format based on DICOM for Analytical Cytology Data ", *Cytometry* 54A pp. 56-65 (2003).
11. R. C. Leif and S. B. Leif, "Evolution of Flow Cytometry Standard, FCS3.0, into a DICOM-Compatible Format". in *Optical Diagnostics of Biological Fluids and Advanced Techniques in Analytical Cytology*, Ed. A. V. Priezzhev, T. Asakura, and R. C. Leif. A. Katzir Series Editor, Progress Biomedical Optics Series, SPIE Proceedings Series, Vol. 2982, pp 354-366 (1997).
12. R.C. Leif, S.H. Leif, S.B. Leif, "CytometryML, a markup language for analytical cytology", in *Manipulation and Analysis of Biomolecules, Cells and Tissues*, D. V. Nicolau, J. Enderlein, and R. C. Leif, Editors, SPIE Proceedings Vol. 4962 pp 288-297 (2003).

13. “XML Schema Part 2: Datatypes Second Edition”, W3C Recommendation 28 October 2004 (http://www.w3.org/TR/2004/REC-xmlschema-2-20041028/).
14. Pricilla Warmsley, “Definitive XML Schema, Prentice Hall”, http://www.phptr.com (2002).
15. “Digital Imaging and Communications in Medicine (DICOM) Part 3: Information Object Definitions”, PS 3.3-2006, National Electrical Manufacturers Association, NEMA, (http://medical.nema.org/dicom/2006/) (2006).
16. B.W. Boehm, “A spiral model of software development and enhancement”, *IEEE Computer*, pp. 61-72 (1988).
17. Digital Imaging and Communications in Medicine (DICOM) Part 16: Content Mapping Resource, PS 3.16-2006, National Electrical Manufacturers Association, NEMA, (http://medical.nema.org/dicom/2006/) (2006).
18. Data Standards Task Force, Bioinformatics Standards for Flow Cytometry Consortium: “Proposal for International Society for Analytical Cytology (ISAC), Gating-ML: Draft Standard for Gating in Flow Cytometry, version 1.1”; http://flowcyt.sourceforge.net/gating/ (2006)
19. Data Standards Task Force, Bioinformatics Standards for Flow Cytometry Consortium: “Proposal for International Society for Analytical Cytology (ISAC), Transformation-ML: Draft Standard for Transformation Description in Flow Cytometry, version 1.0”; http://flowcyt.sourceforge.net/transformation/ (2006).
20. Data Standards Task Force, Bioinformatics Standards for Flow Cytometry Consortium:” Proposal for International Society for Analytical Cytology (ISAC), Compensation-ML: Draft Standard for Compensation Description in Flow Cytometry, version 1.0”; http://flowcyt.sourceforge.net/compensation/ (2006).
21. “RDF Vocabulary Description Language 1.0: RDF Schema, W3C Recommendation 10 February 2004” (http://www.w3.org/TR/2004/REC-rdf-schema-20040210/).
22. R. Brinkman, J. Spidlen, and the Bioinformatics Standards for Flow Cytometry Consortium, “FlowRDF: A Proposal for Describing Flow Cytometry Metadata Using the Resource Description Framework”, http://flowcyt.sourceforge.net/flowrdf/ (2006)
23. “RDF Semantics W3C Recommendation 10 February 2004”, (http://www.w3.org/TR/2004/REC-rdf-mt-20040210/).
24. “XML Schema Datatypes in RDF and OWL, W3C Working Group Note 14 March 2006” (http://www.w3.org/TR/2006/NOTE-swbp-xsch-datatypes-20060314/).
25. J. Spidlen, R. C. Gentleman, P. D. Haaland, M. Langille, N. Le Meur, M. F. Ochs, C. Schmitt, C. A. Smith, A. S. Treister, and R. R. Brinkman, “Data Standards for Flow Cytometry, *OMICS A Journal of Integrative Biology,* Vol. 10, pp. 209-214 (2006).
26. L. C. Seamer, C. B. Bagwell, L. Barden, D. Redelman, G. C. Salzman, J. C. Wood, R. F. Murphy, “Proposed new data file standard for flow cytometry”, version FCS 3.0. *Cytometry* **28**, pp. 118–122 1997.
27. “FCS, Flow Cytometry Standard”, http://www.isac-net.org/ Then search for FCS.
28. M. Birbeck, “RDFA: The Easy Way to Publish Your Metadata, XTech 2006: *“Building Web 2.0”* — 16-19 May 2006, Amsterdam, The Netherlands, http://xtech06.usefulinc.com/schedule/paper/58 (2006).
29. “Semantic Web Best Practices and Deployment Working Group, part of the W3C Semantic Web Activity, XML Schema Datatypes in RDF and OWL”, W3C Working Group Note 14 March 2006, http://www.w3.org/TR/swbp-xsch-datatypes/.