SAND2020-3536C

Exploration of a Data-Enhanced Calibration Certificate as Part of a Complete Measurement Information Infrastructure

Speaker: Collin Delker
Authors: Collin Delker, Mike Roberts, Amaru Robinson, Otis Solomon
Primary Standards Lab
Sandia National Laboratories
P.O. Box 5800, MS 0667, Albuquerque, New Mexico, 87185-0667 USA
Phone: 505-845-7431 Fax: 844-4327 Email: cjdelke@sandia.gov

Abstract

Every calibration laboratory creates calibration certificates that report technical and operational information about a device. Frequently, calibration certificates are issued in Portable Document Format (PDF) with little concern over whether recipients can electronically extract and use the calibration data at their facility. Sometimes data is saved as an image within the PDF forcing the use of optical character recognition or manual transcription to extract any information. These practices effectively lock the data and make it difficult to extract automatically. Without accessible data, tasks involving multiple certificates, such as control charting or interval analysis, become impossible. A universal Measurement Information Infrastructure (MII) includes a calibration certificate in a standardized, open format that allows easy access to the data for analysis, yet can be presented in a traditional, readable form. This paper explores some proof-of-concept ideas under investigation at the Primary Standards Laboratory for such an enhanced calibration certificate. An open specification based on mature technology will ease the transition from existing information systems to new MII standards. This paper describes how to embed Extensible Markup Language (XML) data into a PDF certificate, extract the information for reuse, store calibration certificates in XML format, and extend and customize the certificate to satisfy all requirements in ISO/IEC 17025:2017(E).

1. Introduction

Calibration is a data-driven discipline. Calibration certificates provide data about a device's measured values, uncertainties, tolerance limits, and other performance information. The collected and reported certificate data, including current and historical calibration certificates, establishes whether the equipment is reporting valid measurements, how its uncertainty propagates down the traceability chain, and verification that the equipment is able to serve its intended purpose. Tasks such as control charting and interval analysis require complete historical data from certificates to identify drift or other changes in a device's performance and to assess optimal calibration intervals.

Unfortunately, traditional calibration certificates are often produced in formats that leave the data difficult to access. Most certificates are provided in Portable Document Format (PDF), a file format developed to lay out printed documents for publication. While PDF certificates may look

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

nice, the information in the document is organized in a way optimized for printing, not necessarily for storing and retrieving data [1]. The files lack a standard underlying data structure compatible with data reuse. It is difficult to extract the actual numeric data, whether through software or manual copy and paste, due to inconsistency in the layouts of tables and rows. Even worse, some certificates are rendered as digital images inside the PDF, meaning the numeric data has been effectively translated into pixels rather than actual numbers. In this case, manual transcription or some type of artificial intelligence with optical character recognition are the only options for extracting the values; both are slow processes with a high risk of transcription errors.

While some calibration labs or calibration software systems have databases with readable data formats, they are largely inconsistent and require customized software for every calibration supplier and/or site. The Measurement Information Infrastructure (MII) is a vision for a universal data format in which calibration data can be stored and transmitted electronically without loss of usefulness. Various aspects of MII have been explored [2], [3], [4] with several entities proposing bits and pieces of a complete MII scheme. In a complete MII system, tasks such as validating a calibration provider's scope of accreditation against requirements, performing interval analyses, and performing risk analysis to verify that calibrated equipment can meet its end requirements can all be done easily and automatically.

This paper presents a proof-of-concept method for storing calibration data within a traditional PDF certificate that can be easily extracted manually or with software. Simple software is demonstrated for adding data to and extracting data from the calibration certificate automatically. For users who do not need the raw data, the certificate looks and behaves as any other PDF document, meaning no new software or special workflows are required.

2. Requirements for a Digital Certificate

A useful universal certificate format that overcomes the limitations of many current certificate formats must meet several requirements. Most importantly, it must be able to store all data required by ISO/IEC 17025:2017(E) for the reporting of calibration results [5]. This includes the obvious calibration data, such as measurement results and uncertainties, but also details of the device under test, contact information for the calibration lab, and identification of the calibration method used. Assuming most data formats are capable of storing such information, other requirements for the proposed universal certificate include the following.

Easily Machine-Readable

The motivation for a universal format is the current difficulty of obtaining calibration data for use in further analysis. The new format should allow easy writing and parsing of the data with whatever programming language is deemed most appropriate for the application.

Easily Transformed to Human-Readable Format

Machine readability often means the opposite of human readability. Customers still require traditional printable calibration certificates, so the new format must accommodate them either by adding the ability to transform the machine-readable data into a traditional certificate, or by combining the machine-readable raw data with a traditional certificate.

An Open Format

To be useful and to gain traction around the metrology community, the certificate must be in an open (as in open source) format, with specifications readily available to calibration labs and their customers. The format should not rely on any one commercial entity to provide software for reading or writing certificates (although developing such software could prove a profitable business venture). Rather, the specification should be available for stakeholders to develop their own customized software for creating and working with certificate data.

Useful for All Stages of Calibration Cycle

The format should add value to all stages of the calibration cycle, from the calibration bench where the measurements are made, to the calibration lab's database and recall system, and finally to the customer. Rather than maintaining independent data files or databases for raw data and printable certificates, using a single data file in all stages of this cycle avoids potential data loss due to separation of data from certificate. This single file would be provided directly to customers who could use their own software to read the data.

Flexibility

There are a nearly infinite number of existing calibration certificate layouts and data fields in existence; the proposed MII certificate format must have the ability to support all of them. The data structure should be adapted to suit the existing workflow, not the other way around.

3. The XML-Enhanced PDF Format

While a completely new data specification could be developed, along with custom software for creating and viewing certificates, a much smoother transition and easier widespread adoption is possible by leveraging existing technology, rather than starting from scratch. A promising possibility based on existing technology has been demonstrated at Sandia National Laboratories' Primary Standards Laboratory (PSL). The PSL has experimented with using PDF and embedded Extensible Markup Language (XML) [6], both mature and stable technologies, to provide both a traditional, printed certificate and enhanced information for further data analysis, combined in a single file.

The XML file format has the potential to address the data accessibility concerns without developing a new file specification. All major browsers and operating systems already support viewing XML documents, and most modern programming languages have built-in libraries for reading and writing XML. Software development for calibration applications would simply leverage existing software libraries to read and write specified keys and values into the XML file. XML is a plain-text format, meaning anyone can, in theory, read and write the actual values using a basic text editor.

While an XML file contains the data enclosed in various metadata tags, it can be transformed into a traditional-looking certificate using Extensible Stylesheet Language Transformations (XSLT) [7]. A single XSLT template could be defined for each type of calibration and used to generate PDFs for all calibrations of that type. Alternatively, any software could provide a transformation template for converting raw XML into PDF or any other readable format.

By its very nature (and name), XML is an extensible syntax. If additional information, not defined in the original specification, needs to be recorded, it can easily be added as additional data tags in the relevant section of the file. This addition will not break any software already using the file; if the software does not recognize a tag, it will simply be ignored, but the data remains. There are also mechanisms for storing nontextual data, such as mathematical expressions and images (useful for providing a diagram of sampling locations as specified in ISO 17025, for example), and the files can be digitally signed to ensure data has not been modified since generation.

The PDF, now standardized under ISO 32000-1:2008 [8], has become the standard for sharing documents electronically. Software for viewing and printing PDFs is preinstalled and available in all major operating systems. Since most calibration certificates are already issued as PDFs, it makes sense to continue using this format for the readable part of a calibration certificate. Luckily, the PDF specification allows for embedding data files, invisible to the reader, that can store extra data accessible by software. This approach would allow the PDF file to contain both the traditional printed certificate and the added XML data from the calibration in a single file. Embedded files can be added or extracted using Adobe Reader or other PDF-editing software libraries.

4. Ideal Data Flow

An ideal data flow situation is illustrated in Figure 1. Software on the calibration bench generates the raw data and stores it in a standardized XML format. Additional raw data could be stored externally on a network drive or database. For example, results of every repeated measurement could be saved in CSV files, while the averaged result to appear on the certificate is put in the XML. The measurement data XML is then combined with header information taken from the recall database, including work order numbers and certification and expiration dates. The full XML is then translated into a PDF certificate. This can be done using an XSLT transformation or other software that reads the XML data and formats it into tables in PDF form. The original XML is embedded as a file attachment into the PDF so that the original data is included in the PDF file; then the PDF file stored in the main calibration database. When a certificate is pulled from the database, it appears to most users as an ordinary, printable PDF certificate, but it contains the embedded raw information for users who need it.

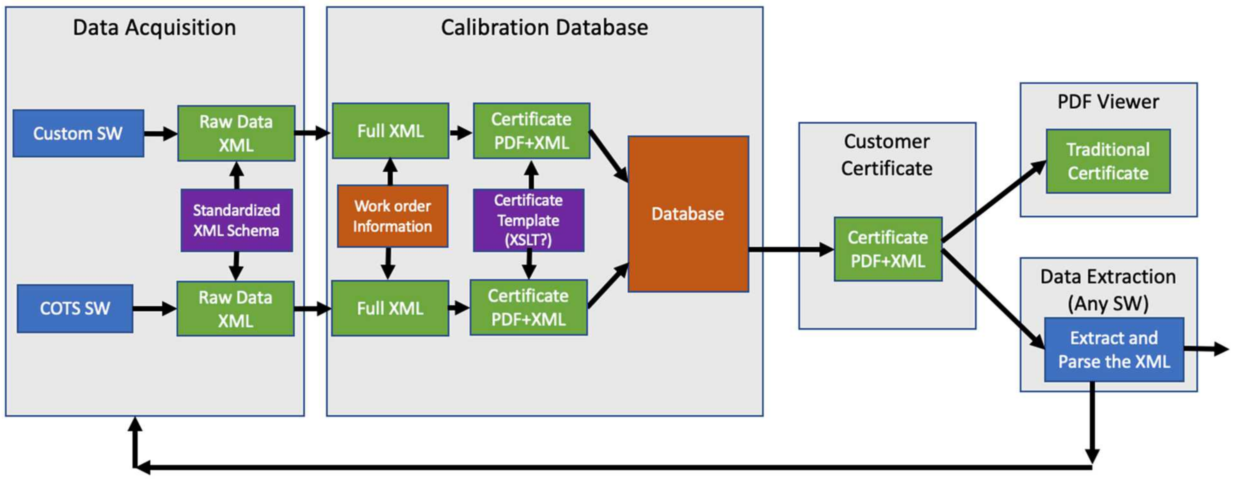

Figure 1. Enhanced PDF certificate data flow.

Such a data flow has several advantages. In addition to customers receiving useful data with their certificate, many manual steps are eliminated, allowing for increased efficiency and reduced potential for errors. All certificates, whether for calibrations or tolerance tests, regardless of what is being calibrated, have similar data schemes.

Of course, fully implementing such a system will not be simple or easy. Reaching consensus on data formats and finding schemes that handle all calibration types and disciplines will be challenging and time consuming. Once a standard is defined, software will need to be updated or created at all stages of the calibration cycle. In transitioning toward such a system, intermediate software tools will be necessary to convert raw data from existing closed-software data into the new format until the old software can be updated or replaced.

5. Proof of Concept Implementation

While challenging from a standardization and consensus standpoint, the actual technological implementation of an XML+PDF certificate system is straightforward due to its reuse of mature technology. A possible XML scheme was developed as described in Appendix A. The scheme was structured to be flexible for encoding any type of calibration data encountered in the PSL, both calibrations that report a value with uncertainty and tolerance tests that provide upper and lower limits with a pass or fail status. The format can contain single-point calibrations, such as calibration of a single length or mass value, or multiple quantity tables for assets such as a digital multimeter with many functions. Eventually, the category names used to specify data in the XML will be updated to use the taxonomy specification under development by the NCSLI committee on MII [9]. An XSLT template used to convert the XML data into a formatted certificate is presented in Appendix B. Additionally, Appendix C provides an example software class, in under 150 lines of Python code, demonstrating how calibration data can be saved to this XML scheme, translated into a PDF using XSLT, embedded into the PDF, and extracted again into useful Python data structures.

For the user of this software class, extracting calibration data becomes a matter of running two lines of code: one to load the certificate into memory, and another to request a particular measurement quantity. Then the raw data can be used for analysis or plotting as applicable. Of course, this demonstration software could be expanded to include additional functionality, such as verifying pass/fail conformity, calculating false accept risk based on the calibration data, or reading multiple certificates to control-chart the results.

```
[15]: cert = Certificate.from_pdf('6688940.pdf')
      cert.get_quantity('DC Volts')
```

[15]:

	Description	Lower Limit	Measured Value	Upper Limit	Units	TUR	%Tol
0	5.000 V	4.994	5.0	5.007	V	>10	0.0
1	55.00 V	54.930	55.0	55.070	V	>10	0.0
2	30.00 V	29.950	30.0	30.050	V	8.1	0.0
3	20.00 V	19.960	20.0	20.040	V	6.6	0.0
4	15.00 V	14.970	15.0	15.030	V	5.8	0.0
5	10.00 V	9.970	10.0	10.030	V	5.0	0.0
6	-10.00 V	-10.030	-10.0	-9.970	V	5.0	0.0
7	-15.00 V	-15.030	-15.0	-14.970	V	5.8	0.0
8	-20.00 V	-20.040	-20.0	-19.960	V	6.6	0.0
9	-30.00 V	-30.050	-30.0	-29.950	V	8.1	0.0
10	-55.00 V	-55.070	-55.0	-54.930	V	>10	0.0
11	300.0 V	299.500	300.0	300.500	V	8.1	0.0
12	1000 V	997.000	1000.0	1004.000	V	6.0	0.0
13	-1000 V	-1004.000	-1000.0	-997.000	V	6.0	0.0

Figure 2. Extracting data from an enhanced PDF certificate for analysis in Python.

MET/CAL [10] is used in some, but not all, of the calibrations done at PSL. Because rigorous international MII standards are not yet in place, MET/CAL does not support the proposed data scheme. However, MET/CAL contains functionality for exporting custom-formatted data reports. As an intermediate step, the PSL has developed software that translates the output of MET/CAL calibration report into the XML+PDF format. A Crystal Report was defined for converting the MET/CAL calibration data into a secondary CSV format, and then a Python code was used to read and convert that data into the desired XML+PDF certificate. Similar transformation software can be easily custom-made for other commercial, off-the-shelf systems.

6. Conclusions and Future Direction

The proof-of-concept enhanced PDF+XML format for certificates is currently being explored at the PSL at Sandia National Laboratories in hopes that these ideas can be considered as part of a broader MII solution for the Nuclear Security Enterprise and the calibration community. Much work needs to be done in refining and standardizing a specification so that all enhanced certificates are universally compatible. A complete MII system would streamline many aspects of calibration data analysis and allow metrologists to use data that was previously inaccessible.

A. XML Certificate Specification

This appendix summarizes the Extensible Markup Language (XML) schema under investigation at the Primary Standards Laboratory (PSL). It was built to handle the various calibration and tolerance test types and be flexible to ensure it can store data from different measurement disciplines from electrical to mechanical.

A.1 Certificate Header Information

Calibration header information, such as details of the device under test, calibration method, calibration conditions, and dates, are given in XML tags `<asset>`, `<customer>`, `<calibration>`, and `<laboratory>`. For example, the `<asset>` section of a multimeter certificate may contain:

```
<asset>
    <name>Multifunction Meter</name>
    <serial>44110519</serial>
    <id>6688940</id>
    <model>179</model>
    <description></description>
    <manufacturer>Fluke Corporation</manufacturer>
    <condition>good</condition>
    <location>SNLNM/TA1/827/136</location>
</asset>
```

The `<calibration>` element contains child elements for describing the calibration method, standards used, and environmental conditions. The `<customer>` and `<laboratory>` elements provide information about who owns the asset and which lab calibrated it.

A.2 Calibration Data

For storing the actual calibration results, PSL's experiments use a set of XML tags defining "quantities" and "measurements". Each certificate may have multiple quantities, such as "AC Volts," "DC Volts," and "Resistance" for a multimeter. Eventually these names will be replaced using the NCSLI's taxonomy definition. Each quantity may contain multiple measurements for various setpoints; for example, a measurement at 10 mV, 100 mV, and 1 V. Other tags under the measurement section depend on the type of calibration.

Example 1: Single Quantity

This example shows the complete data section for a single-length calibration. The value, uncertainty, units, and conformity status are all provided.

```
<quantity name="Length">
    <measurement>
        <value units="nm">181.6</value>
        <uncertainty units="nm">2.8</uncertainty>
        <conformity>pass</conformity>
    </measurement>
</quantity>
```

Example 2: Tolerance Test of Digital Multimeter

This example shows partial data for a tolerance test, which includes the measured value, upper and lower limits, and test uncertainty ratio. A complete data section would contain multiple quantities (e.g. DC Volts, Resistance, DC Current, etc.) and additional measurements under each quantity.

```
<quantity name="AC Volts">
    <measurement>
        <setpoint units="mV">300</setpoint>
        <setpoint units="Hz">45</setpoint>
        <value units="mV">299.9</value>
        <limits>
            <lower units="mV">296.7</lower>
            <upper units="mV">303.3</upper>
        </limits>
        <tur> &gt;10</tur>
        <tolerance>3</tolerance>
    </measurement>
    <measurement>
        <setpoint units="V">5.000</setpoint>
        <setpoint units="Hz">500</setpoint>
        <value units="V">4.995</value>
        <limits>
            <lower units="V">4.947</lower>
            <upper units="V">5.053</upper>
        </limits>
        <tur> &gt;10</tur>
        <tolerance>9</tolerance>
    </measurement>
</quantity>
```

Example 3: Microwave S-Parameter Certificate

Microwave s-parameters require several quantity sections for each s-parameter (S11, S21, etc.) and component (Magnitude, Phase). This example shows the general structure for an S11 Magnitude measurement at two frequencies.

```
<quantity name="S11 Magnitude">
    <measurement>
        <setpoint quantity="frequency" units="MHz">100</setpoint>
        <value units="None">0.2053</value>
        <uncertainty units="None">0.0047</uncertainty>
        <conformity>pass</conformity>
    </measurement>
    <measurement>
        <setpoint quantity="frequency" units="MHz">120</setpoint>
        <value units="None">0.2052</value>
        <uncertainty units="None">0.0047</uncertainty>
        <conformity>pass</conformity>
    </measurement>
</quantity>
```

A.3 Complete Example of an XML Certificate

This example shows a complete XML certificate including header information and calibration results.


```xml
<?xml version="1.0" encoding="UTF-8"?>
<certificate number="6679700_11750243">
    <notes>
        <note>This certificate or report shall not be reproduced except in full, without the advance written approval of the laboratory.</note>
        <note>The as received condition of the standard, set of standards, or measurement equipment described herein was as expected, unless otherwise noted in the body of the certificate or report.</note>
        <note>The results reported above relate only to the items tested or calibrated.</note>
    </notes>
    <asset>
        <name>Standard,Step,Height</name>
        <serial>12392-37-11</serial>
        <id>6679700</id>
        <model>SHS-1800 QC</model>
        <description></description>
        <manufacturer>VLSI Standards, Inc.</manufacturer>
        <condition>good</condition>
        <location>SNLNM/TA1/878/B1600</location>
    </asset>
    <customer>
        <name>PSL-Electrical Lab</name>
        <custodian>Robinson, Amaru</custodian>
        <contact>
            <address>Sandia National Labs</address>
            <phone>505-845-8855</phone>
            <email>pslhelp@sandia.gov</email>
        </contact>
    </customer>
    <calibrationlab>
        <name>Primary Standards Lab</name>
        <description>Sandia National Laboratories</description>
        <department>Length Mass Force Lab</department>
        <contact>
            <address>Albuquerque, New Mexico 87185</address>
            <phone>505-845-8845</phone>
            <email>pslhelp@sandia.gov</email>
        </contact>
    </calibrationlab>
    <calibration>
        <dates>
            <received>2018-10-09</received>
            <calibrated>2018-10-10</calibrated>
            <approved>2018-10-17</approved>
            <expiration>2021-10-16</expiration>
        </dates>
        <method>
            <name>PSL-LMF-CP-4003-001, 02</name>
            <system>MMS 5555</system>
            <description>This step height standard was measured using a coherence scanning interferometer (a specific type of scanning white light interferometer). The calibration results are from an average of 5 individual scans, measurements from each scan are from an average of a minimum of 9 measurements in the marked region of the central step height.</description>
            <traceability>Values and the associated uncertainties reported are traceable to the SI through one of more of the following:
1. Reference standards whose values are disseminated by the PSL and are traceable to National Institute of Standards and Technology (NIST) or, where appropriate, to the national metrological institute of another nation participating in the CIPM MRA;
2. Reference standards whose values are disseminated by a laboratory that has demonstrated competence, measurement capability, and traceability for those values;
3. The accepted value(s) of fundamental physical phenomena (intrinsic standards);
```

```
4. Ratio(s) or other non-maintained standards established by either a self-calibration and/or a direct calibration technique;
5. Standards maintained and disseminated in special cases and where warranted, such as consensus standards where no national or international standards exist.</traceability>
            <coverage>2.0</coverage>
            <decisionrule>The Decision Rule for the As-Found condition is Simple Acceptance, where the measured value is within the previous certification limits.</decisionrule>
        </method>
        <standards>
            <asset>
                <id>6671277</id>
                <description>Measuring,Coordinate,Microscope</description>
                <model>CCI LITE</model>
                <expiration>2020-06-13</expiration>
            </asset>
            <asset>
                <id>6666367</id>
                <description>Standard,Step,Height</description>
                <model>VDS-4.5 QS</model>
                <expiration>2020-10-27</expiration>
            </asset>
            <asset>
                <id>6652235</id>
                <description>Standard,Step,Height</description>
                <model>SHS-9400QC</model>
                <expiration>2022-06-19</expiration>
            </asset>
        </standards>
        <environment>
            <temperature>
                <value unit="C">20</value>
                <uncertainty unit="C">1</uncertainty>
            </temperature>
            <humidity>
                <value unit="%RH">40</value>
                <uncertainty unit="%RH">10%</uncertainty>
            </humidity>
            <pressure>
                <value>NA</value>
            </pressure>
        </environment>
        <authorization>
            <calibrated>Elbara Ziade</calibrated>
            <approved>Andrew Mackrory</approved>
        </authorization>
    </calibration>
    <data>
        <quantity name="Length">
            <measurement>
                <value units="nm">181.6</value>
                <uncertainty units="nm">2.8</uncertainty>
                <conformity>pass</conformity>
            </measurement>
        </quantity>
    </data>
</certificate>
```

B. XSLT Template for Tolerance Test

This XSLT file describes how to translate the above XML data for a tolerance test into a human-readable, formatted, and styled HTML document. The HTML can be converted to PDF using existing methods before embedding the raw XML.

```
<xsl:stylesheet id="stylesheet" version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <html>
            <head>
                <style>
                body {
                  color: #444;
                  font-family: Georgia, Palatino, Times, 'Times New Roman', serif;
                  font-size: 14px;
                  line-height: 1.7;
                  padding: 1em;
                  margin: auto;
                  max-width: 56em;
                  background: #fefefe;
                }
                p {margin: 1em 0;}
                h1, h2, h3, h4, h5, h6 {
                            color: #111;
                            line-height: 125%;
                            margin-top: 2em;
                            font-weight: normal;
                            }
                            h4, h5, h6 {
                  font-weight: bold;
                }
                h1 { font-size: 2.5em; }
                h2 { font-size: 2em; }
                h3 { font-size: 1.5em; }
                h4 { font-size: 1.2em; }
                h5 { font-size: 1em; }
                h6 { font-size: 0.9em; }
                b, strong { font-weight: bold; }
                table {
                  margin-bottom: 2em;
                  border-bottom: 1px solid #ddd;
                  border-right: 1px solid #ddd;
                  border-spacing: 0;
                  border-collapse: collapse;
                }
                table th {
                  padding: .2em 1em;
                  background-color: #eee;
                  border-top: 1px solid #ddd;
                  border-left: 1px solid #ddd;
                }
                table td {
                  padding: .2em 1em;
                  border-top: 1px solid #ddd;
                  border-left: 1px solid #ddd;
                  vertical-align: top;
                }
                </style>
            </head>
            <body>
                <center><h2>Calibration Certificate</h2></center>
                <h3>Item Identification</h3>
                <table>
                        <tr>
                            <td>Asset</td>
                            <td><xsl:value-of select="certificate/asset/id"/></td>
```

```
                        </tr>
                        <tr>
                            <td>Serial</td>
                            <td><xsl:value-of select="certificate/asset/serial"/></td>
                        </tr>
                        <tr>
                            <td>Model</td>
                            <td><xsl:value-of select="certificate/asset/model"/></td>
                        </tr>
                        <tr>
                            <td>Calibration Date</td>
                            <td><xsl:value-of
select="certificate/calibration/dates/calibrated"/></td>
                        </tr>
                </table>
                <h3>Calibration Results</h3>
                <table>
                    <tr>
                        <td><b>Test Description</b></td>
                        <td><b>Lower Limit</b></td>
                        <td><b>Measured Value</b></td>
                        <td><b>Upper Limit</b></td>
                        <td><b>Units</b></td>
                        <td><b>TUR</b></td>
                        <td><b>%Tol</b></td>
                    <td><b>Status</b></td>
                    </tr>
                    <xsl:for-each select="certificate/data/quantity">
                        <tr><td><xsl:value-of select="./@name"/></td></tr>
                        <xsl:choose><xsl:when test="./@range">
                            <tr><td><xsl:value-of select="./@range"/></td></tr>
                        </xsl:when></xsl:choose>
                        <xsl:for-each select="measurement">
                            <tr>
                                <td><xsl:for-each select="setpoint">
                                        <xsl:value-of select="."/>  
                                        <xsl:value-of select="./@units"/>  
                                     </xsl:for-each>
                                </td>
                                <td><xsl:value-of select="limits/lower"/></td>
                                <td><xsl:value-of select="value"/></td>
                                <td><xsl:value-of select="limits/upper"/></td>
                                <td><xsl:value-of select="value/@units"/></td>
                                <td><xsl:value-of select="tur"/></td>
                                <td><xsl:value-of select="tolerance"/></td>
                                <td>
                                    <xsl:choose>
                                        <xsl:when test="value &lt; limits/lower">
                                            <font color="red">FAIL</font>
                                        </xsl:when>
                                        <xsl:when test="value &gt; limits/upper">
                                            <font color="red">FAIL</font>
                                        </xsl:when>
                                        <xsl:otherwise>
                                            PASS
                                        </xsl:otherwise>
                                    </xsl:choose>
                                </td>
                            </tr>
                        </xsl:for-each>
                    </xsl:for-each>
                </table>
                <h3>Authorization</h3>
                <table>
                    <xsl:for-each select="certificate/calibration/authorization/*">
                    <tr>
                        <td><xsl:value-of select="local-name()"/> by</td>
                        <td><xsl:value-of select="."/></td>
                    </tr>
```

```
                    </xsl:for-each>
                </table>
            </body>
        </html>
    </xsl:template>
</xsl:stylesheet>
```

C. Example Code

This Python code can be used to read and write XML-enhanced PDF certificates. The certificate may be created from scratch (i.e. on the calibration bench) given raw calibration data and asset information plus an XSLT template describing how to present the XML data in printable certificate form. Loading certificate data results in a Python/Pandas data table that can be used for subsequent analysis (See Figure 2). In addition to the Pandas library, this code depends on the xlml library for manipulating XML data and translating into HTML, the PyPDF2 library for reading and writing PDF and embedding the XML, and weasyprint for converting the HTML into PDF.

```python
import lxml.etree as ET
from PyPDF2 import PdfFileWriter, PdfFileReader
from weasyprint import HTML as weasyHTML
import pandas as pd

def dict_to_xml(name, items):
    ''' Convert a (possibly nested) dictionary to XML. Each dictionary
        value can be string, list or another dictionary.
    '''
    root = ET.Element(name)
    for item, value in items.items():
        if isinstance(value, dict):
            root.append(dict_to_xml(item, value))
        elif isinstance(value, (list, tuple)):
            for listitem in value:
                root.append(dict_to_xml(item, listitem))
        else:
            ET.SubElement(root, item).text = value
    return root


class Certificate(object):
    ''' XML+PDF Calibration Certificate '''
    def __init__(self, root=None):
        self.root = root if root is not None else ET.Element('certificate')
        if self.root.find('data') is None:
            ET.SubElement(self.root, 'data')

    def __str__(self):
        return ET.tostring(self.root, pretty_print=True).decode()

    @classmethod
    def from_dict(cls, items):
        ''' Create a certificate from a dictionary '''
        root = dict_to_xml('certificate', items)
        return cls(root)

    @classmethod
    def from_xmlfile(cls, fname):
        ''' Load certificate from XML file '''
        tree = ET.parse(fname)
        return cls(tree.getroot())

    @classmethod
    def from_pdf(cls, fname):
        ''' Load certificate from XML embedded in PDF '''
        with open(fname, 'rb') as f:
            reader = PdfFileReader(f)
```

```
            catalog = reader.trailer["/Root"]
            fileNames = catalog['/Names']['/EmbeddedFiles']['/Names']
            attachments = {}
            for f in fileNames:
                if isinstance(f, str):
                    name = f
                    dataIndex = fileNames.index(f) + 1
                    fDict = fileNames[dataIndex].getObject()
                    fData = fDict['/EF']['/F'].getData()
                    attachments[name] = fData
            cert = attachments.get('certificate.xml')
            if cert is None:
                raise ValueError('PDF does not contain embedded XML.')
        return cls(ET.fromstring(cert))

    def to_html(self, template):
        ''' Convert XML to HTML using the XSLT template '''
        transform = ET.XSLT(ET.parse(template))
        return ET.tostring(transform(cert.root)).decode()

    def save_pdf(self, fname, template):
        ''' Save to enhanced PDF with embedded XML, using XSLT template '''
        html = self.to_html(template)
        tree = ET.ElementTree(self.root)
        xml = ET.tostring(tree, pretty_print=True,
                          xml_declaration=True, encoding='utf-8')
        weasyHTML(string=html).write_pdf(fname)
        wPdf = PdfFileReader(fname)
        outPdf = PdfFileWriter()
        outPdf.appendPagesFromReader(wPdf)
        outPdf.addAttachment('certificate.xml', xml)
        with open(fname, 'wb') as f:
            outPdf.write(f)

    def save_xml(self, fname):
        ''' Save the XML data to a file '''
        tree = ET.ElementTree(self.root)
        with open(fname, 'wb') as f:
            f.write(ET.tostring(tree, pretty_print=True,
                                xml_declaration=True, encoding='utf-8'))

    def add_measurement(self, quantity, value, uncertainty=None,
                        LL=None, UL=None, tur=None, tolerance=None,
                        setpoints=None, units=None, uncunits=None):
        ''' Add a measurement to the certificate '''
        data = self.root.find('data')
        qtyelm = cert.root.xpath(f"//data/quantity[@name='{quantity}']")
        qtyelm = qtyelm[0] if len(qtyelm) > 0 else ET.SubElement(data, 'quantity', name=quantity)
        elm = ET.SubElement(qtyelm, 'measurement')

        ET.SubElement(elm, 'value', units=str(units)).text = str(value)
        if uncertainty:
            ET.SubElement(elm, 'uncertainty', units=str(uncunits)).text = str(uncertainty)
        if LL or UL:
            lim = ET.SubElement(elm, 'limits')
            ET.SubElement(lim, 'lower', units=str(units)).text = str(LL)
            ET.SubElement(lim, 'upper', units=str(units)).text = str(UL)
        if tur:
            ET.SubElement(elm, 'tur').text = str(tur)
        if tolerance:
            ET.SubElement(elm, 'tolerance').text = str(tolerance)
        if setpoints:
            for setval, setunit in setpoints:
                ET.SubElement(elm, 'setpoint', units=str(setunit)).text = str(setval)

    def _get_value(self, elm):
        ''' Get dictionary (possibly nested) of values given the Element '''
        values = {}
        if len(elm) == 0 and elm.text is not None:
            values[elm.tag] = elm.text
        for item in list(elm):
            if len(list(item)) > 0:
                values[item.tag] = self._get_value(item)
            else:
                values[item.tag] = item.text  # NOTE: will overwrite duplicates!
        return values
```

```python
    def get_value(self, tag):
        ''' Get a dictionary (possibly nested) of values given the tag name '''
        elm = self.root.find(tag)
        if elm is not None:
            return self._get_value(elm)
        return {}

    def get_quantity(self, name):
        ''' Get DataFrame containing all measurements on a quantity.
            Currently works for tolerance-test certificates to make a table
            with upper/lower limit columns.
        '''
        idx = self.root.xpath('data/quantity/@name').index(name)
        quantity = self.root.xpath('data/quantity')[idx]
        rows = []
        for measurement in quantity:
            setpts = measurement.findall('setpoint')
            setpts = ' '.join(['{} {}'.format(pt.text, pt.attrib.get('units', '')) for pt in setpts])
            valueelement = measurement.find('value')
            LL = float(measurement.find('limits/lower').text)
            UL = float(measurement.find('limits/upper').text)
            units = valueelement.attrib.get('units', '') if valueelement is not None else ''
            value = float(valueelement.text)
            row = [setpts,
                   float(measurement.findtext('limits/lower')),
                   float(measurement.findtext('value')),
                   float(measurement.findtext('limits/upper')),
                   units,
                   measurement.findtext('tur'),
                   float(measurement.findtext('tolerance'))]
            rows.append(row)

        df = pd.DataFrame(rows, columns=['Description', 'Lower Limit',
                                         'Measured Value', 'Upper Limit',
                                         'Units', 'TUR', '%Tol'])
        return df
```

D. References

[1] Adobe Systems Incorporated, "Document management - Portable document format - Part 1: PDF 1.7," 2008.

[2] D. Zajac, "Creating a standardized schema for representing ISO/IEC 17025 scope of accreditations in XML data," in *NCLSI Workshop and Symposium*, 2016.

[3] M. J. Kuster, "Toward a measurement information infrastructure," in *MSA Conference*, Queenstown, New Zealand, 2015.

[4] M. Johnston and A. Chapman, "A proposal for a standard calibration data format," in *Measurement Science Conference*, 2018.

[5] ISO/IEC 17025:2017, "General requirements for the competence of testing and calibration laboratories.," 2017.

[6] W3C, "Extensible Markup Language (XML), Fifth Edition," 26 November 2008. [Online]. Available: http://www.w3.org/TR/xml/.

[7] W3C, "XSL Transformations (XSLT) Version 3.0," 8 June 2017. [Online]. Available: https://www.w3.org/TR/xslt-30/.

[8] ISO 32000-1:2008, "Portable Document Format," ISO, 2008.

[9] C. Walker, M. Kuster and M. Schwartz, "Taxonomy for a Measurement Information Infrastructure," in *Measurement Science Conference*, 2018.

[10] Fluke Calibration, "MET/CAL Software," [Online]. Available: https://us.flukecal.com/products/calibration-software/met-cal®-software. [Accessed 2020].

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.