#22

IMEKO TC8, TC11 and TC24 Joint Conference
October 11-13, 2023, Madeira, Portugal

VALIDATION METHODS IN THE PREPARATION OF XML-BASED DIGITAL CALIBRATION CERTIFICATES (DCC)

G. Söylev Öktem[1]/Presenter, S. Hackel[2], B. Gloger[3], J. Loewe[4], S. Schönhals[5]

[1] Physikalisch-Technische Bundesanstalt (PTB), Braunschweig, Germany, gamze.soeylev-oektem@ptb.de
[2] Physikalisch-Technische Bundesanstalt (PTB), Braunschweig, Germany, siegfried.hackel@ptb.de
[3] Physikalisch-Technische Bundesanstalt (PTB), Braunschweig, Germany, benjamin.gloger@ptb.de
[4] Physikalisch-Technische Bundesanstalt (PTB), Braunschweig, Germany, jan.loewe@ptb.de
[5] Physikalisch-Technische Bundesanstalt (PTB), Braunschweig, Germany, shanna.schoenhals@ptb.de

Abstract:
Calibration certificates are the heart of metrological quality infrastructure and a central element of traceability of measurements. The Digital Calibration Certificate (DCC) is an endeavor to digitalize calibration certificates, developed coordinated by the PTB with the contributions of national and international partners. The DCC is implemented in XML format, and an XSD Schema is developed and maintained by PTB. The DCC schema allows validating the DCCs. Schematron is a schema language which is used to write logical rules for XML files. The PTB's DCC team has prepared a Schematron validation tool with open-source resources.

Keywords: DCC, Schematron, XML Schema, XPATH

1 INTRODUCTION

Calibration certificates are a fundamental part of metrological quality infrastructure. They document the results of calibrations. Unfortunately, they are still paper based or at least not machine-actionable when issued as simple digital documents (such as Word or PDF files). This can lead to many problems, such as missing possibility of reusing the calibration data, media break and associated errors in the transfer process from certificate to customer database, paper waste, etc. Digitalization of calibration certificates is a very important part of the digitalization efforts in metrology because of the obvious benefits like reducing errors and increasing productivity. One of the leading efforts is being undertaken by the Physikalisch-Technische Bundesanstalt (PTB) in Germany and is called the Digital Calibration Certificate (DCC). In this context, generating error-free DCCs plays a crucial role to achieve the maximum benefit and this aspect will be addressed in this paper.

In this paper, we briefly describe the DCC and touch on possible errors that may arise when using the DCC. We then explain how to validate DCCs. We discuss several options to validate DCCs, focussing specifically on schema validation and validation with Schematron. We show what kind of errors can be detected and prevented by schema validation and through Schematron.

2 DIGITAL CALIBRATION CERTIFICATE (DCC)

The PTB coordinates the development of a standard for digital calibration certificates [1]. While there exist efforts to develop PDF/A-3 based DCCs, PTB provides a DCC schema [2] which is based on the Extensible Markup Language (XML) [3]. There is also a comprehensive documentation for the DCC schema [4]. The DCC schema defines the structure for XML files, so it can be used to validate generated DCCs.

3 HOW TO VALIDATE A DCC?

Even though XML files are human readable, it can be overwhelming for the ones who are not familiar with the format. Moreover, to be able to use the full advantages of digitalization, it makes sense to use automated validation methods to make sure that the DCCs are correctly structured and are free from logical errors. The most common method is to use schema validation. Many types of errors can be caught by this method. For example, the information that must be given in the certificate according to the standard, such as an identifier for the certificate, or the name of the calibration laboratory, can be declared as a mandatory field. In this way, if this information is provided in the certificate, this can be noticed by a schema validation.

Even though there are many types of errors that can be detected by schema validation, there are some limitations of this method, because the purpose of a schema is to define the structure of XML files and not to check whether the data is correct or whether integrity constraints are fulfilled. For instance, one may make it mandatory in the schema to enter begin and end performance dates of the calibration. However, the schema cannot check if the begin performance date is earlier than or the same date as the end performance date. Such a check may be conducted with Schematron. Schematron [5] is a schema language, which is not used for creating schemas, but rather makes assertions for specific contexts.

a. Schema Validation

The DCCs can be validated against the XSD schema provided by PTB. XSD schemas define the structure of the XML files and put constraints. In this section, we focus on what type of errors can be caught with a schema validation.

An XML file is called ‘well-formed’ if its syntax is correct. XML files have the typical tree structure. That means that every XML file must have one root, which is the starting element. Every element must have a starting and ending tag and can have attributes, a text content, or other elements, which are called child elements. Every element has exactly one parent, except for the root element which has no parents. A well-formed XML file can be validated against a schema.

In XML files, the element names can be freely defined in a schema. That is why schema validation allows to check if the element names are written correctly.

XML allows checking the order of child elements, allows multiple elements with the same name, and enables limiting the number of these elements. XML has built-in data types, such as ‘xs:string’, ‘xs:date’, ‘xs:datetime’, ‘xs:boolean’. In addition, regular expressions can be used to enforce more detailed constraints on the elements. Enumeration lists can also be defined to provide the users with a list of elements to choose from. This will help with the error minimization.

Figure 1 shows the results of a schema validation with possible errors that could be caught. Many different XML software offer the schema validation with similar structure. Once errors are detected, they can be corrected subsequently.

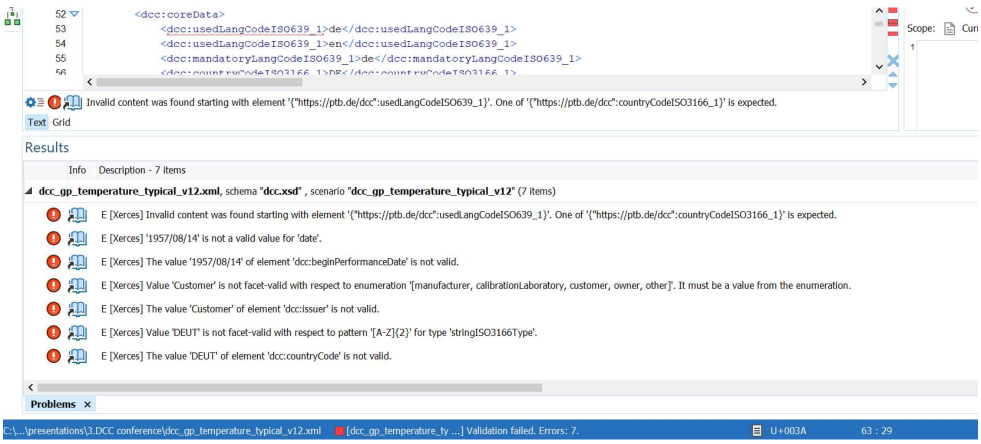

Figure 27: Schema validation results generated with ‘Oxygen XML Demonstrator’ Software

b. Schematron Validation

Schematron is a very simple, and very powerful language. It uses XPATH (XML Path Language) [6] based tests instead of grammar to validate XML files. In other words, a schematron schema does not generate a grammar for XML files, but rather makes assertions for specific context, in other words for specific elements of an XML file. XPATH makes it possible to conduct detailed analyses of XML files [7]. Schematron is an ISO standard. Schematron can be used among other things for constraint checking, naming and design rules, data exploration, and data reporting.

Schematron rules are written for specific contexts of an XML file. There are 6 main elements in schematron. Schematron files have ‘sch’ extension. Figure 2 shows the contents of an example schematron file:

```
1   <sch:schema xmlns="http://purl.oclc.org/dsdl/schematron">
2       <sch:ns uri="https://ptb.de/dcc" prefix="dcc"/>
3       <pattern>
4           <sch:rule context="dcc:content">
5               <sch:assert role="warning" test="@lang">
6                   The language attribute is not used for this element.
7               </sch:assert>
8               <sch:report role="warning" test="@lang">
9                   The language attribute is used for this element.
10              </sch:report>
11          </sch:rule>
12      </pattern>
13  </sch:schema>
```

Figure 28: Schematron example

The rule is written for the context ‘dcc:content’. ‘dcc’ is the prefix for the namespace which is used in the DCC schema. ‘dcc:content’ is an element which is used in many different positions in the DCC schema.

The first line is the starting line of the schematron file. The root element of the schematron files is ‘sch:schema’.

The second line is not a mandatory line. It is used if a namespace is used in contexts to define said namespace. In the case of the DCC, this is ‘dcc’ as prefix and ‘https://ptb.de/dcc’ as namespace, which both are defined originally in the DCC schema.

The rules are written in patterns which is shown on the third line. Patterns can contain multiple rules, and schematron files can contain multiple patterns. If one needs different rules for the same context, they need to be written in different patterns, or else these rules will be ignored.

There are two different ways to apply tests: assert and report. ‘sch:assert’ on the line 5 delivers the message if the tests fails. ‘sch:report’ on the line 8 delivers the message if the test succeeds.

The role of the messages can also be defined by the user with the attribute ‘role’. There are 6 different roles to choose from. For the sake of simplicity, we chose ‘warning’, ‘information’, and ‘error’.

Tests are written using ‘test’ attribute to either ‘sch:assert’ or ‘sch:report’. In this example, there are two different tests for the same context. The messages on the lines 6 and 9, that will be delivered depending on the result of the tests, are user-defined. The two tests for this example are the same. They check the existence of the ‘lang’ attribute of the context ‘dcc:content’. In other words, the message on line 6 will be delivered in case of the absence of the attribute, because the test is attached to an assert. The message on line 9 will be delivered in cases where the ‘dcc:content’ element has the attribute ‘lang’. In both cases, the message will be classified as a ‘warning’.

As shown above, the existence of an attribute can be checked with schematron. Schematron offers the possibility to compare the content of different elements. For instance, it is possible to check that the begin performance date is earlier than or at the same date as the end performance date. The following example in Figure 3 shows measured data from an example DCC:

```
<si:realListXMLList>
        <si:valueXMLList>306.245 373.127 448.249 523.321 593.1510000</si:valueXMLList>
        <si:unitXMLList>\kelvin</si:unitXMLList>
        <si:dateTimeXMLList>1957-08-13T08:15:00Z 1957-08-13T09:15:00Z 1957-08-13T10:15:00Z 1957-08-13T11:15:00Z 1957-08-13T12:15:00Z</si:dateTimeXMLList>
</si:realListXMLList>
```

Figure 29: DCC code example

The ‘si:dateTimeXMLList’ element contains the exact date and time when the measurements were conducted. Hence, there must be the same amount of data in both elements, which is impossible to check with schema validation, but very easy with schematron. It is also possible to compare the number of data given as reference values and measured values, even though they are descendants of different elements.

In DCCs there are some elements which are optional and can be found in different positions in the schema, but a valid certificate must contain this information, such as ambient conditions. The reason for this is that the ambient conditions can be the same for every data point in some certificates, but they also can be different for every data point in a certificate. DCC schema has both options. With schematron, it is possible to check if an ambient condition for every data point in a certificate is declared either globally or locally.

The above examples are general rules that any DCC must conform to. It is also possible to write

DCC specific tests, such as company name or length of data. In the future, PTB will provide an online schematron validation service with the general rules.

Because schematron rules are written for specific XML files, if the schema is changed, the tests might not work for the new XML files. For this reason, it might be necessary to revise existing rules when substantial changes are made to the schema. For instance, the upcoming version 4.0 of the DCC will bring many changes. Version 4.0 will be based on another schema which is called Digital SchemaX [8]. As this version will change the namespace of many elements and probably the structure of some parts of the new schema, all existing schematron rules must be revised or rewritten.

To apply the schematron rules, one can use various commercial tools, such as oXygen, Liquid Software. It is also possible to use free tools or libraries. One possible way is to use Saxon-HE [9] which allows to generate an XSLT file based on the existing schematron file. Then, this XSLT file is applied to the DCC by using a schematron processor [10]. For this, one needs two command line commands:

1. java -jar saxon-he/saxon-he-10.3.jar -s:dcc.sch -xsl:schxslt/core/src/main/resources/xslt/2.0/compile-for-svrl.xsl -o:dcc.xsl
2. java -jar saxon-he/saxon-he-10.3.jar -s:dcc_gp_temperature_simplified_v13.xml -xsl:dcc.xsl -o:result.xml

With this method, one receives an XML file (result.xml in the above example) and must extract the useful information by themselves. It is also possible to create a software. For instance, there is a Python library called 'saxonche', which uses SaxonC with a Java virtual machine. The figure 3 shows the software that is being developed by PTB using open-source resources as a demonstrator:

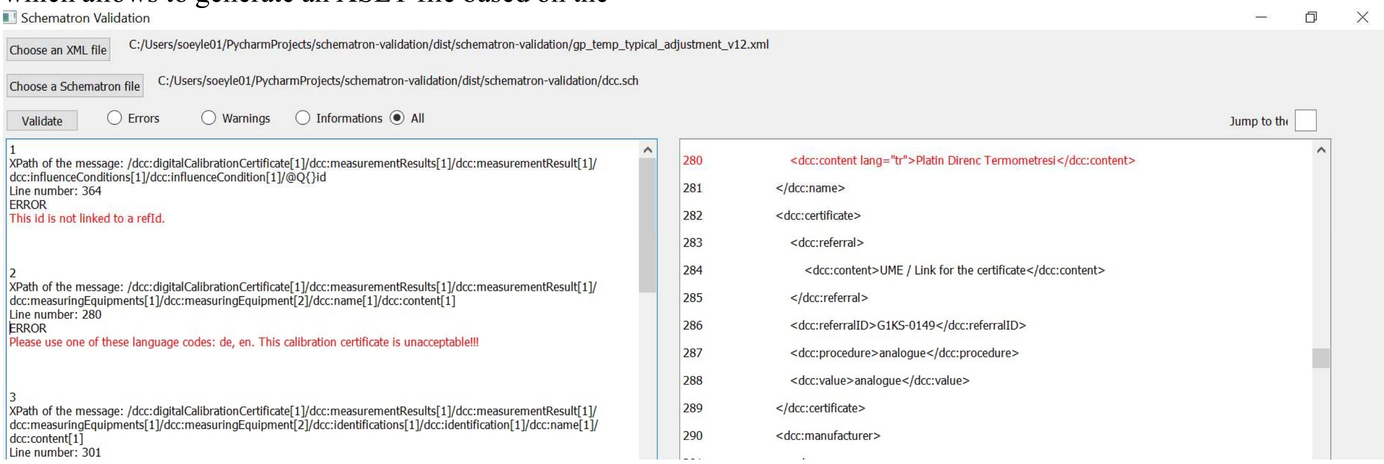

Figure 30: PTB schematron demonstrator

4 SUMMARY

The DCC provides many opportunities. However, it is crucial to generate errors free DCCs. There are different methods to manage that. First and foremost, schema validation must be used. Unfortunately, it is impossible to detect all possible errors. Thankfully, there are additional methods, such as schematron. Schematron is a very easy and elegant way to catch additional errors.

5 ACKNOWLEDGEMENT

The Project GEMIMEG II is carried out with the Grant reference of GEMIMEG 01 MT20001E from BMWK (Federal Ministry for Economic Affairs and Climate Action).

6 REFERENCES

S. Hackel et al., "The fundamental architecture of the DCC", Meas. Sens., vol. 18, p. 100354, Dec. 2021 DOI:10.1016/j.measen.2021.100354.

PTB, "DCC Schema". Online [Accessed 20230618]: https://ptb.de/dcc/v3.2.1/dcc.xsd

Extensible Markup Language (XML) 1.0 (fifth edition). Online [Accessed 20230618]: https://www.w3.org/TR/xml/

PTB, "DCC Documentation". Online [Accessed 20230618]: https://dccwiki.ptb.de/en/home

Schematron. Online [Accessed 20230618]: https://www.xml.com/pub/a/2003/11/12/schematron.html

XPATH. Online [Accessed 20230618]: https://www.w3.org/TR/1999/REC-xpath-19991116/

Schematron. Online [Accessed 20230618]: https://www.data2type.de/xml-xslt-xslfo/schematron/schematron-einfuehrung

G. Söylev Öktem et al. "Digital SchemaX and the Future of the Digital Calibration Certificate". Online [Accessed 20230830]: https://www.m4dconf2022.ptb.de/fileadmin/documents/m4dconf2022/Material/Paper/IMEKOTC6-M4Dconf-2022-P50-SOELEV-OEKTEM-et-al.pdf

Saxon-HE. Online [Accessed 20230618]: https://www.saxonica.com/html/download/java.html

Schematron Processor. Online [Accessed 20230618]: https://github.com/schxslt/schxslt