Edinburgh Research Explorer

Tracking and viewing modifications in digital calibration certificates

Tracking and viewing modifications in digital calibration certificates

Vashti Galpin[1], Ian Smith[2], Jean-Laurent Hippolyte[2]

[1] *School of Informatics, University of Edinburgh, United Kingdom*
[2] *National Physical Laboratory, United Kingdom*

ABSTRACT
Trust in current and historical calibration data is crucial. The recently proposed XML schema for digital calibration certificates (DCCs) provides machine-readability and a common exchange format to enhance this trust. We present a prototype web application developed in the programming language Links for storing and displaying a DCC using a relational database. In particular, we leverage the temporal database features that Links provides to capture different versions of a certificate and inspect differences between versions. The prototype is the starting point for developing software to support DCCs and the data with which they are populated and has underlined that DCCs are the tip of the iceberg in automating the management of digital calibration data, activity that includes data provenance and tracking of modifications.

1. INTRODUCTION

Information relating to the calibration of an instrument or artefact is captured in documents referred to as calibration certificates. These documents are typically provided either as physical paper-based documents or in electronic form, for example, in archivable Portable Document Format (PDF-A). A downside of both approaches is that the information they contain is not machine-readable, that is it is not available in a form that can be read and processed by computer. Using the information therefore requires, to some degree, human involvement and as such is prone to error, for example, arising from the transcription of numerical information from paper to computer.

Recent initiatives have considered the replacement of paper-based and electronic calibration certificates by fully machine-readable calibration certificates, referred to as "digital calibration certificates", often abbreviated to "DCCs".

The European Metrology Programme for Innovation and Research (EMPIR) [1] funded the "SmartCom" Joint Research Project (2018-2021) [2] which developed a framework for DCCs.

The framework allows the key calibration certificate components of administrative data and measurement results to be stored. The recipient of a DCC will have, or be able to develop, software to ingest and use the information stored therein.

Ensuring the integrity and longevity of current and historical calibration data is essential for calibration laboratories to maintain long-term trusted relationships with their customers. It is also a requirement of ISO/IEC 17025 accreditation for these laboratories [3], along with the preservation of information about the methods, processes, software, equipment, and staff involved in the calibration task. This accompanying information, which is essential to trace and potentially repeat the conditions under which the calibration is performed, is more generally known as "provenance information" [4]. Laboratories must be able to track and compare the differences between successive calibrations of the same artefact in data (for example, newly observed measurement results) or in provenance (for example, a different employee now signs off the calibration results). Since DCCs are containers for the trusted exchange of calibration data, their potential to increase the productivity of laboratories relies on

automated, integrated, and time-based curation of this data and associated provenance.

The structure of this paper is as follows: we provide background on DCCs before we go on to describe our prototype. We discuss the choices made in the development of the prototype and illustrate its usage with a screenshot. We conclude with ideas for further work. This paper is an extended version of an earlier conference paper [5].

2. DIGITAL CALIBRATION CERTIFICATES

The structure of a DCC [6] reflects the information that is required by ISO/IEC 17025 [3] for reporting calibration results. A DCC comprises two compulsory sections. The first compulsory section contains the administrative information including that which is generally presented on the first page of a paper-based certificate. The second compulsory section contains measurement results. The measurement results section itself relies upon a framework, referred to as the "D-SI", developed specifically for the storage of measurement data. The framework ensures that quantity values, units of measurement and uncertainty information can all be represented. A DCC may also include two optional sections. The first optional section contains information presented purely for humans, for example, calibration-specific data sheets, and other auxiliary, machine-interpretable data, for example, relational tables. The second optional section contains an encoded version of a human-readable version of the certificate.

The D-SI and DCC frameworks specify, for example, how numerical values and date and time information should be presented and uses the BIPM SI brochure [7] and the siunitx package for LaTeX [8] as the basis for the provision of units of measurement.

The D-SI and the DCC may be implemented in a language chosen by the user, for example, Extensible Markup Language (XML) or JavaScript Object Notation (JSON). The SmartCom project has developed and made available an XML schema for the D-SI framework [9] while the Physikalisch-Technische Bundesanstalt (PTB), the National Metrology Institute (NMI) of Germany, has made available an XML schema for the DCC [10].

Demonstrators have been developed by NMIs using Excel and Python [11], [12] and as a web application (GEMIMEG) [13] to illustrate the use of these schema. A different approach considers embedding calibration data in PDF calibration certificates to support machine-readability [14]. NMIs have also been investigating good practice for DCCs [15] and usage in specific domains [16], [17]. Developers of commercial software for calibration are also starting to support the use of DCCs [18], [19]. The first IMEKO TC6 M4DConf, held in Berlin in 2022, focussed on the digital transformation of metrology and many submissions considered DCCs [20].

We are aware that the frameworks (and related schemas) may be subject to future updates, for example, if the BIPM SI brochure is updated, or if the requirements for the contents of a calibration certificate change. In the approach we have taken, dealing with XML schema changes is straightforward, and in fact occurred during our development when a newer version of the DCC schema became available, and we discovered that the dummy certificate with which we were working was missing a compulsory XML element.
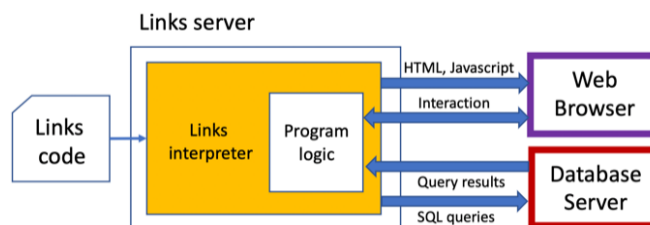

Figure 1. The Links architecture.

3. THE DCC PROTOYTPE

Our prototype application brings together concepts from metrology (specifically, the XML schema for DCCs) and academic research in databases and programming languages and enables the transfer of research ideas from academia. On the computer science research side, there are two distinct concepts: temporal databases and a technique for storing XML documents in a relational database, that are combined using the Links programming language, a tierless language that supports language-integrated query and provides a single language for the development of web applications with database back-ends [21], [22]. We now consider these three components in detail: Links, temporal databases, and storage of XML documents; before considering some details of the implementation of the application.

3.1. The Links Programming Language

Links is a strongly-typed and statically-typed functional programming language, that is *cross-tier* (or *tierless*): it removes the need for the developer to write JavaScript or a particular database query language (in this case, PostgreSQL [23]) for the different tiers of a web application providing data from a database. In particular, Links assists in the writing of correct software by providing *language-integrated query* which allows the programmer to write high-level queries that are transformed to correct SQL queries with known performance for execution on the database. Furthermore, the type-checking of variables and functions before program execution reduces the likelihood of run-time errors.

Figure 1 illustrates the Links architecture. The Links interpreter takes the Links code and generates correct HTML and JavaScript to run in the user's browser, and to support exchange of data between the browser and Links program, thus allowing an interactive webpage. The interpreter also generates correct SQL queries which are passed to a database server for execution which responds with the results. Using this approach, data intensive operations can take place on the database server, using the fact that database systems such as PostgreSQL employ efficient data structures and algorithms for queries. In particular, indexes on key values can make search operations very efficient.

Links provides syntax to specify database tables as well as comprehension syntax to iterate over each row of a database table. We use the MVU (Model-View-Update) library [24] for the development of the web interface as it provides functions to use the element of the Bootstrap toolkit for website development [25], thereby affording a higher level of abstraction by which to achieve an appealing web frontend.
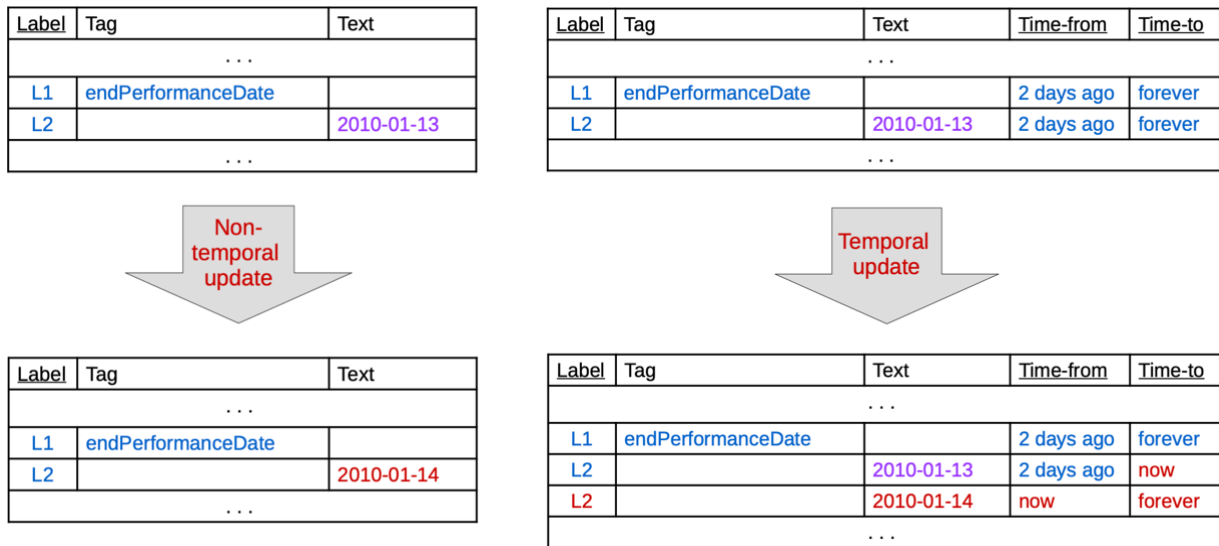
Figure 2. Non-temporal update of a field versus a temporal update.

```
<dcc:digitalCalibrationCertificate
   xmlns:dcc="https://ptb.de/dcc"  xmlns:si="https://ptb.de/si"
   schemaVersion="3.1.2" ... >
 <dcc:administrativeData>
  <dcc:dccSoftware> ... </dcc:dccSoftware>
  <dcc:coreData>
   <dcc:countryCodeISO3166_1>GB</dcc:countryCodeISO3166_1>
   <dcc:usedLangCodeISO639_1>en</dcc:usedLangCodeISO639_1>
    ...
   <dcc:performanceLocation>laboratory</dcc:performanceLocation>
  </dcc:coreData>
  <dcc:items>
   <dcc:item>
    <dcc:name>
     <dcc:content>Digital thermometer</dcc:content>
    </dcc:name>
    ...
   </dcc:item>
  </dcc:items>
  <dcc:calibrationLaboratory> ... </dcc:calibrationLaboratory>
  <dcc:respPersons> ... </dcc:respPersons>
  <dcc:customer> ... </dcc:customer>
  <dcc:statements> ... </dcc:statements>
 </dcc:administrativeData>
 <dcc:measurementResults>
  ...
 </dcc:measurementResults>
</dcc:digitalCalibrationCertificate>
```

Figure 3. An example DCC in XML: text nodes are shown by white text on a dark blue background, attributes by black text on a grey background and the other components are XML elements defined by opening and closing tags. The ellipses indicate where elements have been omitted.

3.2. Temporal databases

Temporal databases capture *when a piece of data is valid*. In the case of a Relational DataBase Management System (RDBMS), we are interested in the time period for which a row in a database table is valid, and the inclusion of additional fields (columns) allows this time period to be recorded [26], [27]. We are interested in the provenance of DCCs and hence we want to capture when data changes in the database, namely *transaction-time* information. Temporal databases can also record *valid-time* information which is information about when something is true in the real world.

Figure 2 demonstrates the difference between an update in a standard relational database versus one in a temporal database supporting transaction time. In the standard case, the new data overwrites the old data and there is no record of the fact that there was a previous value for the field. In a temporal database, two additional fields capture information about when the data was changed, permitting queries such as "What were the values on this day last year?", "Has this field ever had a different value?", and "Which fields have the most frequent changes?". In the example, the date fields are given informally; however, in the database, the fields are of SQL data type TIMESTAMP WITH TIME ZONE and cannot be NULL. In terms of performance in the DCC scenario, since updates are not frequent occurrences, the additional database operations required in the temporal case will not impose a significant overhead.

Links has recently been extended with temporal features that have been demonstrated in the context of curated scientific databases [28] and we use these features in the development of our prototype.

3.3. XML documents

For this paper, we have used a synthetic DCC in XML, containing dummy administrative data and measurement results, formatted according to the XML schema [10] made available by PTB.

XML is a tree-structured language, and an XML document is an (inverted) tree of nodes starting from a root node. Each node can have any number of child nodes. Nodes without children are called leaves.

Each node in an XML document is either an element or a text node (a string of characters). An element consists of an opening

tag and closing tag (or a combined opening and closing tag). The child nodes of the element (if any) appear in between a pair of opening and closing tags of that element. These child nodes can be elements themselves or text (strings of characters). A text node has no children (and hence is a leaf).

An XML schema describes the tags that can be used in a document and how they can be nested [29]. Additionally, opening tags may contain attributes. Example nodes and attributes are given in Table 1 and Table 2. Figure 3 provides an example DCC using the XML schema. In the example, <dcc:performanceLocation> is an opening tag and </dcc:performanceLocation> is a closing tag, together providing an XML element, which contains the text node laboratory. In the figure, the opening tag for the dcc:digitalCalibrationCertificate element contains three attributes which are shown and these attributes provide sources for the XML schema and give the schema version used in the document. The children of this initial element are the elements dcc:administrativeData and dcc:measurementResults.

The temporal features of Links are currently supported by an RDBMS, hence it is necessary to map from the XML schema to this database. There were two choices here: either to use the DCC XML schema to create the appropriate tables in the RDBMS or to treat the data in a schema-agnostic manner and capture it as an XML document. We choose the latter approach because changes to the XML schema will not require changes to the RDBMS schema.

The Dynamic Dewey (DDE) labelling scheme supports the labelling of XML documents. It uniquely labels each node in an XML document and thereby provides a key for storing the node in a relational table. It has good query and insertion performance [30]. It is an extension of the Dewey labelling scheme which has good query performance but differs from it by it removing the need for relabelling (which reduces update performance) when the XML document is updated because Dynamic Dewey can provide unique labels for new nodes.

To store an XML document using this approach, only two tables are required [30] (although a third table can be added to store path information separately – we omit these details here). This approach requires the generation of a unique value to serve as a key for each node (an element or a text node) in an XML document. The DDE algorithm determines this unique label (referred to as dde) which consists of a sequence of integers (the nodes can be labelled with negative numbers in certain cases) separated by dots – so-called dot decimal notation. A total order is defined for the labels (which differs from the standard ordering over dot-decimal notation), and this ordering can be used to reconstruct the nesting of the XML document as well as to determine relationships between nodes such as parent, child, and sibling.

The main table stores nodes and is similar to the one illustrated in the right-hand side of Figure 2. Its schema is (dde, tag, text, path, tt-from, tt-to). Each row has the dde value for the node as the key attribute. If the node is an element, the element tag will appear in the tag column, otherwise if it is a text node, its text string will appear in the text column (one of these two columns is always empty). The path column stores the list of the element tags followed from the root element to the parent of the node. The last two columns tt-from and tt-to store the timestamps that describe when the node is valid. Because the dde value captures the relationships between nodes, there is no need to explicitly refer to other nodes in a row of this table.

The second table contains attribute information and has the schema (dde, attr, value, tt-from, tt-to). In an element, each attribute can only appear at most once, hence the attribute name and the dde value are sufficient to uniquely identify the attribute's value.

Consider the following XML fragment:

<el1 attr1="A value">
  <el2>Text<\el2>
  <el3 attr2="Another value"><\el3>
<\el1>.

The DDE algorithm will generate the content shown in Tables 1 and 2. For this example, the length of each segment of the key is 3 characters. In the implementation, this length is a parameter, and it can be set appropriately for document size.

As can be seen from this explanation, the actual details of the XML schema are not used for the database schema – there is no table for customer details, for example.

By using the DDE ordering with the appropriate Links code to specify joins over the relations, documents can be stored, retrieved and displayed, as shown by our implementation.

3.4. Implementation

The prototype provides an interface for viewing a single DCC. Straightforward extensions include the ability to edit the DCC and to validate that it adheres to the DCC and D-SI XML schemas. Further obvious extensions are the ability to work with many DCCs, to compare different DCCs, and to support the signing of DCCs. We are interested in provenance beyond tracking changes in the data, such as recording which person and what software made the changes.

In the longer term, we will require systems that support the whole calibration workflow including automated collection and processing of machine-readable calibration data.

Using the interface, it is possible to view the current state of the document, the state at a previously specified date and time, and to compare two versions at two specified dates and times. It is possible to hide or show specific subtrees of the document, and when doing the comparison, to see the subtrees where changes have been made.

The Change Analysis option allows the user to see a history of all changes over the life of a DCC. Figure 4 shows how changes are highlighted as well as cumulative totals of insertions and changes for each tag and text node in the document. There are various options to view subtrees for a specific tab or to control the display of the whole document. The figure also shows how details of changes are displayed. (Note that this screenshot is of a different DCC to the one presented in Figure 3.) The Links code for the prototype is available at https://github.com/vcgalpin/dcc-xml-temporal [31].

Table 1. Example node table.

dde	tag	text	path	tt-from	tt-to
001	el1		\	...	...
001.001	el2		\el1\	...	...
001.001.001		Text	\el1\el2\	...	...
001.002	el3		\el1\	...	...

Table 2. Example attribute table.

dde	attr	value	tt-from	tt-to
001	attr1	A value	...	...
001.002	attr2	Another value	...	...

As mentioned above, a more complete application should be able to validate against the DCC and D-SI XML schemas. The level of validation that is provided by these schemas is very high but not complete. In some cases, patterns are used to ensure that values are within bounds. For example, the si:probabilityValueType [9] has the following associated pattern

```
<xs:pattern value="\+?((0(\.\d*)?)|(1(\.0*)?))"/>.
```

This pattern specifies that a value of this type is an element of the interval [0,1], as required for a probability value. However, not all aspects of the input for a DCC are specified in this manner and it would be useful to be able to check for correct spelling of measurement units and probability distribution names, as well as requirements that are specific to a particular domain of measurement or instrument. Good practice guides for DCCs in specific domains are being prepared [14] and another possibility is to provide additional schemas that prescribe requirements for specific groups of instruments.

Moreover, not all functionality needs to be supported directly in Links. For example, schema validation, which could be provided by different software such as the TraCIM system 2.0 developed by PTB as part of the SmartCom project [32]. The application validates XML documents for the D-SI schema, and also undertakes additional compliance checking of the units used and the minimum metrological information required by VIM [33] and GUM [34]. However, the provision of a suitable interface for users will require the integration of the different pieces of software.

3.5. Discussion

Considering our choice to work with a relational database, Links may in future support the storage of XML documents directly with temporal features, and hence the translation would no longer be necessary. In that case, the schema be used more directly to store data, such as the customer details, for example.

Since the DCC is only one component of the envisioned digital calibration workflow and can be seen as output and data exchange from some form of data storage, there is no need to use an XML storage format. Dealing with temporal data in a relational database is better understood and less experimental so we believe it makes sense to use a RDBMS for a system for digital
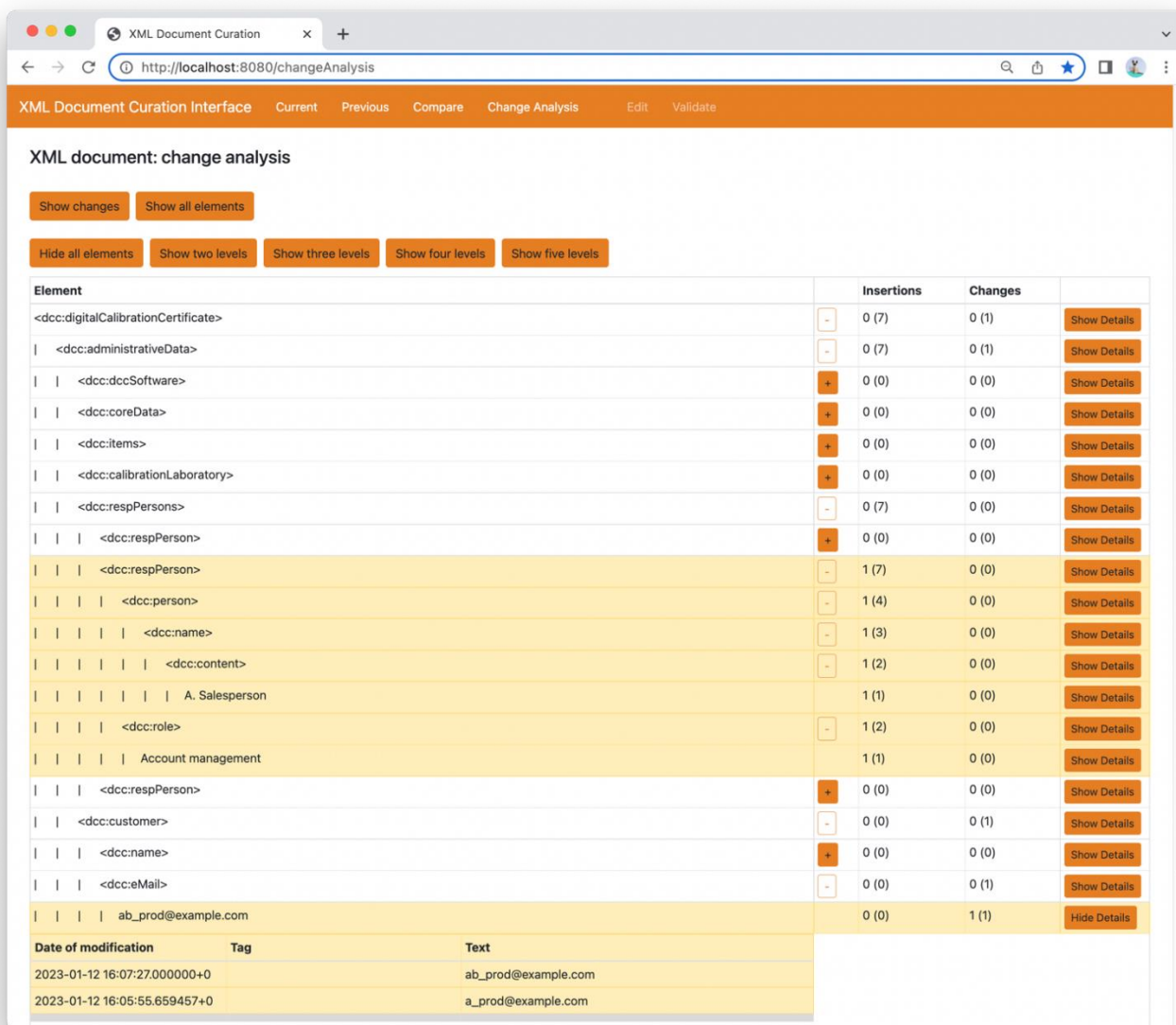

Figure 4. A screenshot of the Change Analysis interface showing an insertion and a change of email – both are highlighted. The insertions and changes columns provide information about the number of modifications. The first number shows the number of changes for the particular tag or text node, and the number in brackets shows the number of changes that have happened below and including the current tag or text node.

calibration data. Working in a schema-less fashion with the XML document provides the ability to cope with changes in the DCC XML schema without affecting the storage in the RDBMS, and we have already found this approach to be of benefit when changes were made to the DCC schema.

Another approach to deal with update provenance is provided by databases that support versioning such as OrpheusDB [35]. This relational database is based on the concepts that are used in software versioning systems such as git [36] and is motivated by the requirement to track changes in curated scientific databases. Temporal databases can be seen as providing a linear model of time whereas versioning supports a non-linear model of time. In the non-linear model, it is possible for separate branches to have different updates that then must be merged together at a later time point. It seems unlikely that DCCs require this level of complexity. Moreover, versioning usually requires users to explicitly create versions, whereas temporal database provide an automatic approach to tracking changes. Furthermore, querying in the linear model is more straightforward than in the non-linear model.

4. FUTURE WORK AND CONCLUSIONS

The prototype we have developed demonstrates that the combination of DCCs and temporal databases results in streamlined curation of calibration data. The prototype leverages features of the Links language

(a) to facilitate the development of a user-friendly web front-end,

(b) to integrate relational database queries on a database that maps with the DCC XML tree structure, and

(c) to enable automated tracking of changes made to calibration records over time, as well as the explanation and timeline for those changes.

Looking to the future, further advances in the area of calibration certificates are in the pipeline, such as Digital SchemaX [37] which will include Envelope Digital Certificates for grouping related calibration certificates together, as well as schema for digital reference materials and digital test certificates, and these advances will require appropriate digital solutions.

Another possibility that results from the digital storage of multiple certificates is to link information across certificates, and to develop new aspects of traceability, and related to this development, certificate validity and frequency of calibration. In many cases, NMIs and calibration laboratories might not know the exact needs of the users at the end of the measurement chain. To meet those needs, a comprehensive traceability chain including the specification of the dependence between consecutive measurement stages is therefore required [38]. Our temporal approach to the curation of calibration data could be used to retain detailed information that the end-users would want to process in a way that was not anticipated by the NMI and calibration laboratories, for example, analysing the risks associated with factors that impact multiple stages across the measurement chain.

To conclude, DCCs are the tip of the iceberg (Figure 5) when considering the whole lifecycle of machine-readable calibration data. Their creation and use depend on software to support storage and provenance of the underlying calibration data. The development of the prototype has raised interesting questions about what such a system should be able to provide, and we will continue with our investigations and development.

Figure 5. DCCs are the tip of iceberg: software support is needed for the full lifecycle of machine-readable calibration data. (Iceberg image designed by pikisuperstar / freepik)

REFERENCES

[1] EURAMET, European Metrology Programme for Innovation and Research (EMPIR). Online [Accessed 12 April 2022] https://www.euramet.org/research-innovation/research-empir/

[2] PTB, SmartCom Project website. Online [Accessed 12 April 2022] https://www.ptb.de/empir2018/smartcom/home/

[3] ISO/IEC 17025:2017; General Requirements for the Competence of Testing and Calibration Laboratories. International Organization for Standardization and International Electrotechnical Commission: Geneva, Switzerland, 2017. Online [Accessed 28 March 2023] https://www.iso.org/publication/PUB100424.html

[4] J. Cheney, A. Chapman, J. Davidson, A. Forbes, Data provenance, curation and quality in metrology, in Advanced Mathematical and Computational Tools in Metrology and Testing XII. World Scientific, 2022, pp. 167–187. DOI: 10.1142/9789811242380_0009

[5] V. Galpin, I. Smith, J.-L. Hippolyte, Supporting provenance of digital calibration certificates with temporal databases, IMEKO TC6 Int. Conf. on Metrology and Digital Transformation (M4Dconf 2022), Berlin, Germany, 19-21 September 2022, 4 pp. DOI: 10.21014/tc6-2022.024

[6] S. G. Hackel, F. Härtig, J. Hornig, T. Wiedenhöfer, T. Schrader, A. Scheibner, J. Loewe, L. Doering, B. Gloger, J. Jagieniak, D. Hutzschenreuter, G. Söylev-Öktem, The fundamental architecture of the DCC, Measurement: Sensors, 18, 2021, 100354. DOI: 10.1016/j.measen.2021.100354

[7] BIPM, The International System of Units (SI), 9th ed.; Bureau International des Poids et Mesures: Sèvres, France, 2019. Online [Accessed 12 April 2022] https://www.bipm.org/documents/20126/41483022/SI-Brochure-9-EN.pdf

[8] Comprehensive TeX Archive Network, siunitx: A comprehensive (SI) units package. Online [Accessed 12 April 2022]
https://ctan.org/pkg/siunitx

[9] PTB, Schema for Digital System of Units (D-SI). Online [Accessed 1 July 2022]
https://gitlab1.ptb.de/d-ptb/d-si/xsd-d-si

[10] PTB, Schema documentation for dcc.xsd, Online [Accessed 12 April 2022]
https://www.ptb.de/dcc/v3.1.0/autogenerated-docs/doku_3.1.0_all.html

[11] T. Bruns, J. Nordholz, D. Röske, T. Schrader, A demonstrator for measurement workflows using digital calibration certificates (DCCs), Measurement: Sensors, vol. 18, December 2021, 100208.
DOI: 10.1016/j.measen.2021.100208.

[12] D. Röske, A visual tool for generating digital calibration certificates (DCCs) in Excel, Measurement: Sensors, vol. 18, December 2021, 100175.
DOI: 10.1016/j.measen.2021.100175

[13] The GEMIMEG Tool. Online [Accessed 11 November 2022])
https://www.gemimeg.ptb.de/gemimeg-tool/#/

[14] G. Boschung, M. Wollensack, M. Zeier, C. Blaser, C. Hof, M, Stathis, P. Blattner, F. Stuker, N. Basic, F. Grasso Toro, PDF/A-3 solution for digital calibration certificates, Measurement: Sensors, vol. 18, December 2021, 100282.
DOI: j.measen.2021.100282

[15] B. Gloger, L. Doering, S. Hackel, J. Jagieniak, G. S. Öktem, Input management for the DCC, IMEKO TC6 Int. Conf. on Metrology and Digital Transformation (M4Dconf 2022), Berlin, Germany, 19-21 September 2022, 4 pp.
DOI: 10.21014/tc6-2022.034

[16] T. Mustapää, P. Nikander, D. Hutzschenreuter, R. Viitala, Metrological challenges in collaborative sensing: applicability of digital calibration certificates, Sensors 2020, 20, 4730,
DOI: 10.3390/s20174730

[17] I. Smith, Y. Luo, D. Hutzschenreuter, The storage within digital calibration certificates of uncertainty information obtained using a Monte Carlo method, Metrology 2022, 2(1), 33-45.
DOI: 10.3390/metrology2010003

[18] J. Nummiluikki, S. Saxholm, A Kärkkäinen, S, Koskinen, Developing and testing digital Calibration Certificates in an industrial application, IMEKO TC6 Int. Conf. on Metrology and Digital Transformation (M4Dconf 2022), Berlin, Germany, 19-21 September 2022, 4 pp.
DOI: 10.21014/tc6-2022.026

[19] M. Nicklich, M. Mende, From digital device under test to digital calibration certificate - challenges and solutions for the calibration of measuring instruments and sensors in the digital future, IMEKO TC6 Int. Conf. on Metrology and Digital Transformation (M4Dconf 2022), Berlin, Germany, 19-21 September 2022, 4 pp.
DOI: 10.21014/tc6-2022.020

[20] IMEKO TC6 International Conference on Metrology and Digital Transformation (M4Dconf 2022), Berlin, Germany, 19-21 September 2022. Online [Accessed 18 November 2022]
https://www.m4dconf2022.ptb.de/home

[21] E. Cooper, S. Lindley, P. Wadler, J. Yallop, Links: Web programming without tiers, 5th Int. Symposium on Formal Methods for Components and Objects (FMCO 2006), LNCS 4709, 2006.
DOI: 10.1007/978-3-540-74792-5_12

[22] The Links team, Links: Linking Theory to Practice for the Web. Online [Accessed 12 April 2022]
https://links-lang.org.

[23] The PostgreSQL Global Development Group, PostgreSQL: The World's Most Advanced Open Source Relational Database. Online [Accessed 12 April 2022]
https://www.postgresql.org/

[24] S. Fowler, Model-View-Update-Communicate: session types meet the Elm architecture, 34th European Conf. on Object-Oriented Programming (ECOOP 2020), Virtual, 15 - 17 November 2020.
DOI: 10.4230/LIPIcs.ECOOP.2020.14

[25] Bootstrap Team, Build fast, responsive sites with Bootstrap. Online [Accessed 20 November 2022]
https://getbootstrap.com/

[26] C. Jensen, R. Snodgrass, Temporal database. Encyclopedia of Database Systems. Springer, 2009.
DOI: 10.1007/978-0-387-39940-9_395

[27] R. Snodgrass, Developing Time-Oriented Database Applications in SQL, Morgan Kaufmann, 1999.

[28] S. Fowler, V. Galpin, J. Cheney, Language-integrated query for temporal data, Proc. of the 21st ACM SIGPLAN Int. Conf. on Generative Programming: Concepts and Experiences (GPCE '22), Auckland, New Zealand, 6 - 7 December 2022, pp. 5-19.
DOI: 10.1145/3564719.3568690

[29] World Wide Web Consortium, XML Technology. Online [Accessed 12 April 2022]
https://www.w3.org/standards/xml/

[30] L. Xu, T.W. Ling, H. Wu, Z. Bao, DDE: From Dewey to a fully dynamic XML labeling scheme, ACM SIGMOD Int. Conf. on Management of Data (SIGMOD 2009), Providence Rhode Island, USA, 29 June - 2 July2009, pp. 719-730.
DOI: 10.1145/1559845.1559921

[31] V. Galpin, vcgalpin/dcc-xml-temporal: v0.1.1, Zenodo, 2023.
DOI: 10.5281/zenodo.7551691

[32] I. Smith, Y. Lou, L. Heindorf, B. Müller, D. Hutzschenreuter, S. Schönhals, Good practice guides SmartCom validation, Zenodo, 2020.
DOI: 10.5281/zenodo.3816696

[33] JCGM 200:2012 International Vocabulary of Metrology – Basic and general concepts and associated terms (VIM), 2012. Online [Accessed 28 March 2023]
https://www.bipm.org/documents/20126/2071204/JCGM_200_2012.pdf

[34] JCGM 100:2008 Evaluation of Measurement Data – Guide to the expression of uncertainty in measurement (GUM), 2008. Online [Accessed 28 March 2023]
https://www.bipm.org/documents/20126/2071204/JCGM_100_2008_E.pdf

[35] S. Huang, L. Xu, J. Liu, A.J Elmore, A. Parameswaran, ORPHEUSDB: bolt-on versioning for relational databases, The VLDB Journal 29, 2020, 509–538.
DOI: 10.1007/s00778-019-00594-5

[36] Git community, git: open source distributed version control system. Online [Accessed 18 January 2023]
https://git-scm.com

[37] G. S. Öktem, S. Hackel, F. Härtig, J. Loewe, B. Gloger, J. Jagieniak, Digital SchemaX and the future of the digital calibration certificate, IMEKO TC6 Int. Conf. on Metrology and Digital Transformation (M4Dconf 2022), Berlin, Germany, 19-21 September 2022, 4 pp.
DOI: 10.21014/tc6-2022.028

[38] B. D. Hall, D. R. White, Digital representation of measurement uncertainty, Measurement: Sensors, vol. 18, December 2021, 100074.
DOI: 10.1016/j.measen.2021.100074