NPL REPORT MS 56

CASE STUDY ON THE GENERATION AND USE OF DIGITAL CALIBRATION CERTIFICATES

I M SMITH, R I VELTCHEVA AND D J GRIFFIN

JUNE 2024

Case study on the generation and use of digital calibration certificates

I M Smith
Data Science department, NPL

R I Veltcheva
Thermal and Radiometric Metrology department, NPL

D J Griffin
National Composites Centre

ISSN 1754-2960

https://doi.org/10.47120/npl.MS56

National Physical Laboratory
Hampton Road, Teddington, Middlesex, TW11 0LW

This work was funded by the UK Government's Department for Science, Innovation & Technology through the UK's National Measurement System programmes.

Approved on behalf of NPLML by
Louise Wright, Head of Science, Data Science department.

CONTENTS

1 BACKGROUND

Compared to other aspects of metrology, the dissemination of calibration information has undergone only relatively modest digital modernisation. Many calibration services continue to disseminate calibration information via calibration certificates printed on paper. Other services have been updated to deliver calibration certificates taking the form of electronic files, for example, digitally signed Portable Document Format (PDF) documents. Both paper and electronic calibration certificates suffer from the same issue, namely that human effort is required to transfer the information they contain to somewhere, such as a spreadsheet, where it can be processed by computer. Such a process is time-consuming and, more significantly, prone to error.

The European Metrology Programme for Innovation and Research (EMPIR) funded the Joint Research Project “Communication and validation of smart data in IoT-networks” (short name “SmartCom”) [1] that ran from 2018–2021 and considered how paper and electronic calibration certificates could be replaced by fully machine-readable certificates. The project delivered a framework for what are referred to as “digital calibration certificates”, or simply “DCCs”. The DCC framework itself relies on a framework for the provision of measurement data, a fundamental component of a calibration certificate. This underpinning framework is referred to as the “Digital SI”, or simply “D-SI”. The outputs of the SmartCom project, including implementations of the D-SI and DCC frameworks, were made freely available [1]. Since the end of the project, updates have been and continue to be made to both frameworks [2].

Effort is currently underway to better understand how existing calibration services can be updated to generate and deliver, in a secure manner, digitally signed DCCs. Such modernisation brings a number of challenges. For example:

- Providers of calibration services need to incorporate DCC generation into those services.
- Users of calibration services need to integrate DCCs into their existing workflows.

In an attempt to begin to address both challenges, the National Physical Laboratory (NPL) [3] and the National Composites Centre (NCC) [4] undertook the joint case study summarised in this report. The report is organised as follows. Section 2 introduces the case study and its aims. Section 3 outlines the calibrations undertaken and section 4 describes how DCCs are subsequently generated, both of these activities taking place at NPL. Section 5 discusses the management and potential for exploitation of DCCs by NCC. Concluding remarks are presented in section 6.

2 OVERVIEW OF THE CASE STUDY

Prior to the case study, NPL and NCC each possessed different levels of awareness of DCCs:

- NPL had been a partner in the SmartCom project and therefore was fully aware of DCCs. However, during the project effort had primarily been focused on the development of the D-SI and DCC frameworks and there had been little time available to consider application of the frameworks to real world calibrations.
- NCC were aware of the outputs of the SmartCom project and keen to learn more about DCCs.

NPL and NCC therefore identified that a joint case study would provide a means for both organisations to increase their general understanding of DCCs and how the D-SI and DCC frameworks can be implemented in practice. In particular:

- NPL could gain experience in a number of DCC-related aspects:
  - The storage of information to allow efficient DCC generation.
  - The development of software for DCC generation.
- NCC could begin to investigate:
  - The potential options for DCC storage.
  - The automated use of the information stored in a DCC.

Initial discussion between NPL and NCC involved identifying the metrology area in which to work. For several reasons, it was decided to focus on the calibration of temperature sensors:

- Information from such calibrations is used directly by NCC.
- The volume of measurement data obtained in such calibrations is relatively small.
- The availability of guidance material for DCCs in the temperature area on the DCC Wiki area [2].

The additional decision was made that the case study should concentrate solely on the information provided within a DCC. While a number of issues such as DCC signing and secure DCC provision need to be addressed by NPL, it was felt that a longer timescale was required to identify, investigate and implement approaches addressing such issues, and that restricting the scope of the case study to information only would be most beneficial.

The case study involved three distinct activities:

1. NPL undertook calibration of three temperature sensors (section 3).
2. NPL developed software to generate a DCC for each calibration and delivered DCCs to NCC (section 4).
3. NCC investigated the management and potential for use of the DCCs (section 5).

3 CALIBRATION OF TEMPERATURE SENSORS

Three temperature sensors were provided by NCC for calibration in terms of the International Temperature Scale of 1990 (ITS-90). This section presents information about each sensor, the calibration method and a summary of the calibration results.

3.1 SENSOR 1

Sensor details

Digital indicator with probe, model TLC1598, manufactured by EBRO, serial number 002-3118 and measurement resolution 0.1 °C.

Calibration method

The probe was immersed directly in a Jofra RTC-700 dry block calibrator to a depth of 100 mm. The sensor was calibrated by comparison of the displayed temperatures from the test thermometer with those of the laboratory reference thermometer at a series of steady temperatures over the range 0 °C to 150 °C and the results given apply to the combined performance.

Throughout the calibration the ambient temperature was between 20 °C and 24 °C.

Figure 1 shows the calibration set-up.

Figure 1 Calibration set-up for the EBRO sensor

Results
The results of the calibration are set out in Table 1, where the corrections to obtain the ITS-90 temperature are stated to the nearest 0.1 °C and represent the means of a set of observations. To obtain the correct temperature, the appropriate correction from the calibration table should be added to the reading, with due regard to the sign.

Table 1 Calibration results for the EBRO sensor

ITS-90 temperature °C	Measured temperature °C	Correction °C
0.0	0.0	0.0
50.0	49.9	+0.1
100.0	100.0	0.0
150.0	149.9	+0.1
0.0	0.0	0.0

Uncertainty
The reported expanded uncertainty of calibration, at the time of test, is 0.2 °C. This value includes all uncertainties in the measurements and in realising the calibration temperatures and is based on a standard uncertainty multiplied by a coverage factor k = 2, providing a coverage probability of approximately 95 %.

3.2 SENSOR 2
Sensor details
Digital indicator, model 53 II B, manufactured by Fluke, serial number 5370219WS and measurement resolution 0.1 °C.

Calibration method
The digital indicator was connected to a temperature calibrator model Fluke 525B via a type K thermocouple cable. The unit was connected to the thermocouple output/input port of the calibrator. A series of temperatures in the range -190 °C to 1200 °C were simulated on the calibrator and measured by the digital indicator.

Throughout the calibration the ambient temperature was between 20 °C and 24 °C.

Figure 2 shows the calibration set-up.

Figure 2 Calibration setup for the Fluke sensor

Results
The results of the calibration are set out in Table 2, where the corrections to obtain the ITS-90 temperature are stated to the nearest 0.1 °C and represent the means of a set of observations. To obtain the correct temperature the appropriate correction from the calibration table should be added to the reading, with due regard to the sign.

Table 2 Calibration results for the Fluke sensor

Temperature set point °C	Measured temperature °C	Correction °C
-190.0	-189.7	-0.3
-100.0	-99.8	-0.2
-50.0	-49.8	-0.2
-20.0	-19.8	-0.2
0.0	0.2	-0.2
20.0	20.2	-0.2
50.0	50.2	-0.2
100.0	100.2	-0.2
200.0	200.2	-0.2
400.0	400.1	-0.1
600.0	600.0	0.0
800.0	800.0	0.0
1000.0	1000.0	0.0
1200.0	1200.0	0.0

Uncertainty
The reported expanded uncertainty of calibration, at the time of test, is 0.4 °C. This value includes all uncertainties in the measurements and in realising the calibration temperatures and is based on a standard uncertainty multiplied by a coverage factor k = 2, providing a coverage probability of approximately 95 %.

3.3 SENSOR 3

Sensor details

Digital indicator, model PRO1384, manufactured by RS, serial number 191008014 and measurement resolution 0.1 °C.

Calibration method

The digital indicator was connected to a temperature calibrator model Fluke 525B via a type K thermocouple cable. The unit was connected to the thermocouple output/input port of the calibrator. A series of temperatures in the range from -50 °C to 1200 °C were simulated on the calibrator and measured by the digital indicator.

Throughout the calibration the ambient temperature was between 20 °C and 24 °C.

Figure 3 shows the calibration set-up. Note that channel 2 could not be calibrated due to existing damage to the device display.

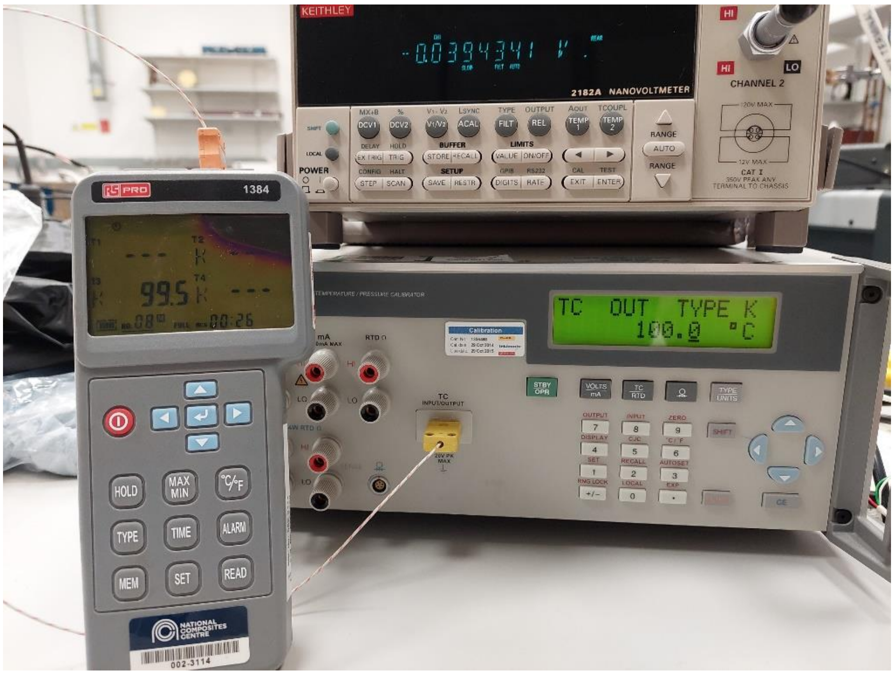

Figure 3 Calibration setup for the RS sensor

Results

The results of the calibrations are set out in Tables 3, 4 and 5 for channels 1, 3 and 4, respectively, where the corrections to obtain the ITS-90 temperature are stated to the nearest 0.1 °C and represent the means of a set of observations. To obtain the correct temperature the appropriate correction from the calibration table should be added to the reading, with due regard to the sign.

Table 3 Calibration results for the RS sensor, channel 1

Calibrator set point °C	Measured temperature °C	Correction °C
-50.0	-49.5	-0.5
0.0	0.4	-0.4
50.0	50.4	-0.4
100.0	100.3	-0.3
500.0	499.8	+0.2
1200.0	1198.9	+1.1

Table 4 Calibration results for the RS sensor, channel 3

Calibrator set point °C	Measured temperature °C	Correction °C
-50.0	-49.6	-0.4
0.0	0.3	-0.3
50.0	50.1	-0.1
100.0	100.1	-0.1
500.0	500.0	0.0
1200.0	1199.4	+0.6

Table 5 Calibration results for the RS sensor, channel 4

Calibrator set point °C	Measured temperature °C	Correction °C
-50.0	-49.8	-0.2
0.0	0.1	-0.1
50.0	49.9	+0.1
100.0	99.8	+0.2
500.0	499.6	+0.4
1200.0	1199.1	+0.9

Uncertainty

The reported expanded uncertainty of calibration, at the time of test is 0.6 °C. This value includes all uncertainties in the measurements and in realising the calibration temperatures and is based on a standard uncertainty multiplied by a coverage factor $k = 2$, providing a coverage probability of approximately 95 %.

4 DCC GENERATION

NPL had previously considered the storage of all necessary calibration information within worksheets in Microsoft Excel workbooks and developed proof-of-concept Python software to extract information from worksheets and generate a DCC. The case study allowed that approach to be implemented for real world calibrations. This section provides a summary of the DCC generation process.

4.1 DCC OVERVIEW

A DCC is essentially a text file that contains information presented in the structured way defined by the DCC and D-SI frameworks. The requirement for structure in a DCC naturally leads to its implementation using data serialisation formats such as Extensible Markup Language (XML) [5] or JavaScript Object Notation (JSON) [6]. Example implementations developed during and following the SmartCom project have used XML, with schemas provided for both the D-SI and DCCs [2]. DCCs developed for the case study have used XML.

An XML DCC[1] has two compulsory sections defined by the elements `dcc:administrativeData` and `dcc:measurementResults`. The former element, as its name suggests, contains administrative data that is typically presented on the first page of a current calibration certificate, and also additional information, such as details of software used to generate the DCC. The latter element contains not only the results of the calibration but additional information, such as influence conditions, relating to the calibration.

An XML DCC may also contain one or both of the optional elements `dcc:comment` and `dcc:document`. The former element contains information, additional to that provided in the element `dcc:measurementResults`, that is specifically intended for humans rather than computers, for example, figures. The latter element contains an encoded human-readable form of all or part of the DCC.

Elements in an XML DCC may have associated attributes that allow more precise properties of those elements to be specified.

For simplicity, both the optional elements of an XML DCC and the use of XML attributes were deemed to be outside the scope of the case study. Instead, they will be considered by NPL as part of future DCC activity.

The structure of the element `dcc:administrativeData` and its child elements is shown in Figure 4. The element contains six compulsory and one optional child elements as follows:

- Element `dcc:dccSoftware` (compulsory) contains information about the software used to generate the DCC.
- Element `dcc:coreData` (compulsory) contains essential information about the calibration.
- Element `dcc:items` (compulsory) contains information that allows identification of the item or items being calibrated.
- Element `dcc:calibrationLaboratory` (compulsory) contains information about the laboratory undertaking the calibration.
- Element `dcc:respPersons` (compulsory) contains information about the person or persons responsible for releasing the DCC.
- Element `dcc:customer` (compulsory) contains information about the customer for whom the calibration is being undertaken.

[1] For the case study, the most recent version (3.1.2) of the DCC framework available at the time was used.

- Element `dcc:statements` (optional) allows statements that apply to the entire calibration process to be stored.

Each child element then contains its own set of child elements.

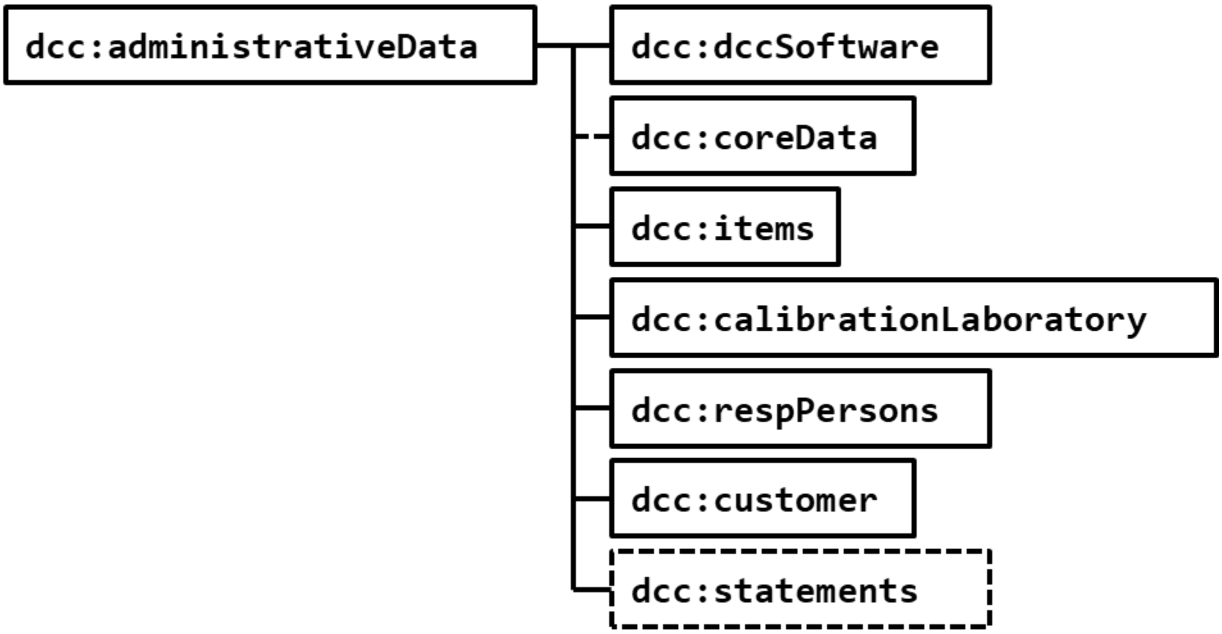

Figure 4 Schematic view of element `dcc:administrativeData` and its child elements in an XML DCC. Dashed lines indicate an optional element.

4.2 INFORMATION STORAGE
For each calibration undertaken, two Microsoft Excel workbooks were created to store information required to populate, respectively, the element `dcc:administrativeData` and element `dcc:measurementResults` of the DCC.

In the “administrative data” workbook, a worksheet was created to store information for each of the child elements of the element `dcc:administrativeData` listed in section 4.1. In the “measurement results” workbook, a single worksheet was created to store the calibration results and other relevant information.

To illustrate the relationship between the child elements of the element `dcc:administrativeData` and the information stored in the corresponding worksheet in the “administrative data” workbook, consider the element `dcc:coreData`.

The structure of the element `dcc:coreData` and its child elements is shown in Figure 5. The element contains seven compulsory and three optional child elements as follows:
- Element `dcc:countryCodeISO3166_1` (compulsory) identifies the country in which the calibration was performed.
- Element `dcc:usedLangCodeISO639_1` (compulsory) identifies the language(s) used in the DCC.
- Element `dcc:mandatoryLangCodeISO639_1` (compulsory) identifies the primary language(s) used in the DCC.
- Element `dcc:uniqueIdentifier` (compulsory) contains a unique identifier for the DCC.
- Element `dcc:identifications` (optional) contains further information to identify the DCC.

- Element `dcc:receiptDate` (optional) identifies the date on which the calibration laboratory received the item(s).
- Element `dcc:beginPerformanceDate` (compulsory) identifies the date on which calibration of the item(s) began.
- Element `dcc:endPerformanceDate` (compulsory) identifies the date on which calibration of the item(s) ended.
- Element `dcc:PerformanceLocation` (compulsory) identifies where the calibration was performed.
- Element `dcc:previousReport` (optional) allows reference to a previous calibration certificate which was replaced by this DCC.

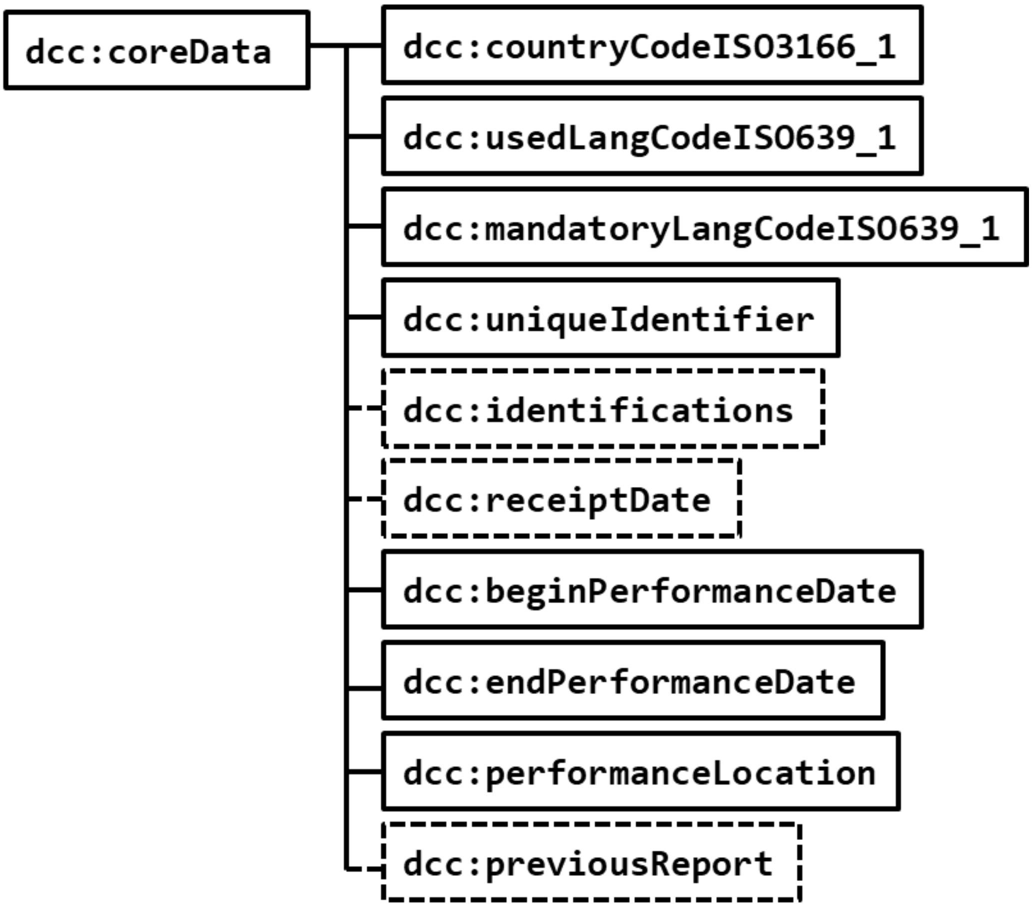

Figure 5 Schematic view of element `dcc:coreData` and its child elements in an XML DCC. Dashed lines indicate an optional element.

Figure 6 shows the contents of the “Core data” worksheet of the “Administrative data” workbook. Information is provided to populate the compulsory child elements of the element `dcc:coreData`. The contents of cells C2 to C8 inform that:

- The calibration was undertaken in the United Kingdom.
- English is the only language used in the DCC.
- English is therefore the primary language of the DCC.
- The unique identifier for the DCC is 2022070197/1.
- The calibration began on 19 July 2022.
- The calibration ended on 19 July 2022.
- The calibration was performed at a branch of the calibration laboratory (the address of which is provided in the “Calibration laboratory” worksheet of the “Administrative data” workbook).

	A	B	C
1			
2		Country code	GB
3		Language code	en
4		Mandatory language code	en
5		Unique identifier	2022070197/1
6		Calibration start date	2022-07-19
7		Calibration end data	2022-07-19
8		Performance location	laboratoryBranch

Figure 6 Contents of the "Core data" worksheet of the "Administrative data" workbook.

A similar approach is used to provide information to populate child elements of the other child elements of element `dcc:administrativeData`.

For the calibration of each of the three temperature sensors described in section 3, "Administrative data" and "Measurement results" workbooks were created.[2]

4.3 SOFTWARE FOR DCC GENERATION

While it is possible to create a DCC manually from scratch by constructing elements, their child elements, their child elements, etc., and copying and pasting information from the worksheets, this approach is not a practical long-term solution. For example, when working with large numbers of measurement results, the transfer of information would be both time-consuming and subject to error.

Generating DCCs using software that requires minimal or no human interaction is the preferred approach. For the case study, the focus was not to develop software that is optimised in terms of time taken to run, efficiency of code, etc., but simply to develop software that works and, if possible, has a degree of reusability, for example, written using modules whose use is not restricted to a single calibration.

The approach implemented involves the use of a template XML DCC, i.e., an XML DCC that contains minimal information but that is still valid according to the D-SI and DCC schemas. This template XML DCC is reproduced in Appendix A.

For each child element of the element `dcc:administrativeData`, information is read from the corresponding worksheet of the "Administrative data" workbook. The contents of the child element are then updated using that information. In a similar manner, information is read from the "Measurement results" workbook and used to populate the element `dcc:measurementResults` of the DCC. Having generated the new DCC, a test is undertaken to ensure that it is valid according to the D-SI and DCC schemas.

The decision was made to use the Python programming language due to:

- Its widespread use nowadays.
- The ease of writing software in a modular way.
- The anticipated availability of libraries to undertake the main tasks, namely reading information from Microsoft Excel workbooks and working with XML. Without

[2] For the case study, the workbooks were created and populated manually, with cell contents either typed in or pasted after being copied from other sources. It is noted that both processes are subject to error. Were this or a similar information storage approach implemented as part of a live calibration service, workbook creation would ideally be automated.

undertaking an exhaustive review of available libraries, the libraries `openpyxl` and `xmlschema` were identified as having the required functionality.

With `xmlschema`, an XML element is encoded to a nested dictionary. Information on the encoding process is provided in Appendix B.

The contents of the element `dcc:coreData` in the template DCC are encoded to the nested dictionary having value

```
{'dcc:countryCodeISO3166_1': 'GB',
 'dcc:usedLangCodeISO639_1': ['en'],
 'dcc:mandatoryLangCodeISO639_1': ['en'],
 'dcc:uniqueIdentifier': 'Unique identifier',
 'dcc:beginPerformanceDate': '1111-12-31',
 'dcc:endPerformanceDate': '1111-12-31',
 'dcc:performanceLocation': 'laboratory'}
```

Information is read in from the “Core data” worksheet of the “Administrative data” workbook and the dictionary values originating in the template are overwritten by this information to give the updated dictionary value

```
{'dcc:countryCodeISO3166_1': 'GB',
 'dcc:usedLangCodeISO639_1': ['en'],
 'dcc:mandatoryLangCodeISO639_1': ['en'],
 'dcc:uniqueIdentifier': '2022070197/1',
 'dcc:beginPerformanceDate': '2022-07-19',
 'dcc:endPerformanceDate': '2022-07-19',
 'dcc:performanceLocation': 'laboratoryBranch'}
```

The updated dictionary value is then decoded to XML.

5 DCC MANAGEMENT AND EXPLOITATION

5.1 DCC MANAGEMENT

NCC currently use the Asset Panda software platform [7] to manage the calibration records of calibrated measurement devices. Asset records on the system contain device identification and ownership details, as well as calibration intervals, date of last calibration and calibration due date; documents such as copies of calibration certificates can also be uploaded and attached. This is a standalone system, on which the details of each device/artefact are managed by the nominated owner.

The existing workflow by which device calibrations and associated records are managed is as follows:

- Owner receives notification from Asset Panda that device/artefact calibration is due.
- Owner arranges calibration with service provider and ships device/artefact.
- Owner sets status on Asset Panda to “Out for Calibration”.
- Service provider performs calibration and ships device back to owner with calibration certificate (in some instances, a PDF certificate is emailed directly to the owner).
- Owner sets status on Asset Panda to “Available”, updates date of last calibration and calibration provider.
- Owner scans calibration certificate (if paper certificate) and uploads a PDF copy of the certificate to Asset Panda.

This workflow relies on the device owner to:

- Cross-check device details between Asset Panda and the certificate.
- Update the device record.
- Upload a certificate copy.

In particular, certificate upload can be overlooked if the owner does not have the certificate to hand in a suitable format when the device record is updated, for instance when updating from a mobile device. For some calibration service providers, the certificate has to be downloaded from a web portal, which can also obstruct this part of the workflow.

Failure to upload a certificate can cause delays when information from the certificate is required for reasons such as confirming the suitability of a device for a particular measurement activity, calculating uncertainty budgets for measurements, etc.

This case study explored how the use of DCCs could enable a certificate-led workflow for the management of calibration records within an organisation, thus mitigating much of the reliance on humans to ensure the accuracy and completeness of records. This was achieved by exploiting the Application Programming Interface (API) of the Asset Panda software platform to programmatically update a device’s record, based on information extracted from the DCC.

A lightweight web application (webapp), the user interface of which is shown in Figure 7, was created to facilitate upload of calibration data in the DCC format to the Microsoft Azure API Management platform[3], see Figure 8, from where API call sequences were managed and executed.

[3] Microsoft Azure is the chosen cloud services provider for NCC.

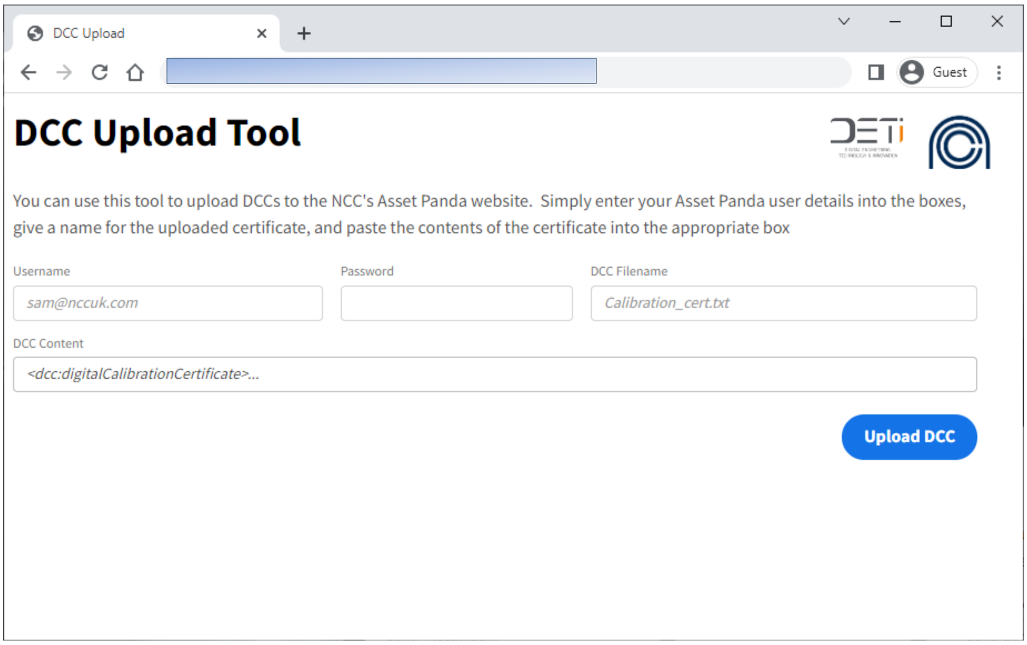

Figure 7 Web application user interface used to upload DCCs to Asset Panda

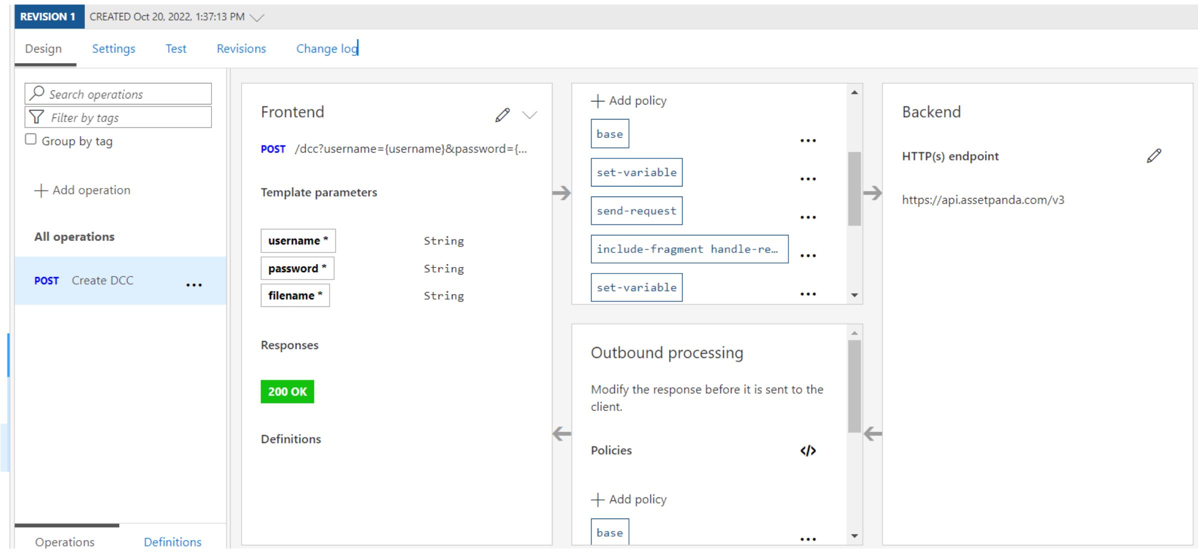

Figure 8 Microsoft Azure API Management platform used to manage and execute API calls to Asset Panda

The steps involved, illustrated as a sequence diagram in Figure 9, can be summarised as follows:

- The user (i.e., device owner) navigates to the webapp URL and enters login credentials, a certificate ID and uploads the XML DCC to the web application by pressing the “Upload DCC” button.
- The customer-specified identifier for the device (i.e., NCC asset ID number) is extracted from the XML DCC and used to identify the corresponding Asset Panda record via the “Asset No.” field.

```
<dcc:administrativeData>
  <dcc:items>
    <dcc:item>
      <dcc:identifications>
        <dcc:identification>
          <dcc:issuer>customer</dcc:issuer>
          <dcc:value>999-9999</dcc:value>
          <dcc:name>
            <dcc:content>Customer asset ID number</dcc:content>
          </dcc:name>
        </dcc:identification>
      </dcc:identifications>
    </dcc:item>
  </dcc:items>
<dcc:administrativeData>
```

- The name of the calibration provider/laboratory, calibration completion date and calibration results are then extracted from the XML and written to the corresponding record fields, namely “Calibration Supplier”, “Last Calibration” and “Calibration Results”.

```
<dcc:administrativeData>
  <dcc:calibrationLaboratory>
    <dcc:contact>
      <dcc:name>
        <dcc:content>NPL</dcc:content>
      </dcc:name>
    </dcc:contact>
  </dcc:calibrationLaboratory>
<dcc:administrativeData>

<dcc:administrativeData>
  <dcc:coreData>
    <dcc:endPerformanceDate>2022-07-19</dcc:endPerformanceDate>
  </dcc:coreData>
</dcc:administrativeData>

<dcc:measurementResults>
  <dcc:measurementResult>
    <dcc:name>
      <dcc:content>Measurement results</dcc:content>
    </dcc:name>
    <dcc:influenceConditions>
    …
    </dcc:influenceConditions>
    <dcc:results>
    …
    </dcc:results>
  </dcc:measurementResult>
</dcc:measurementResults>
```

- Finally, the entire XML is uploaded to the record as a document attachment.

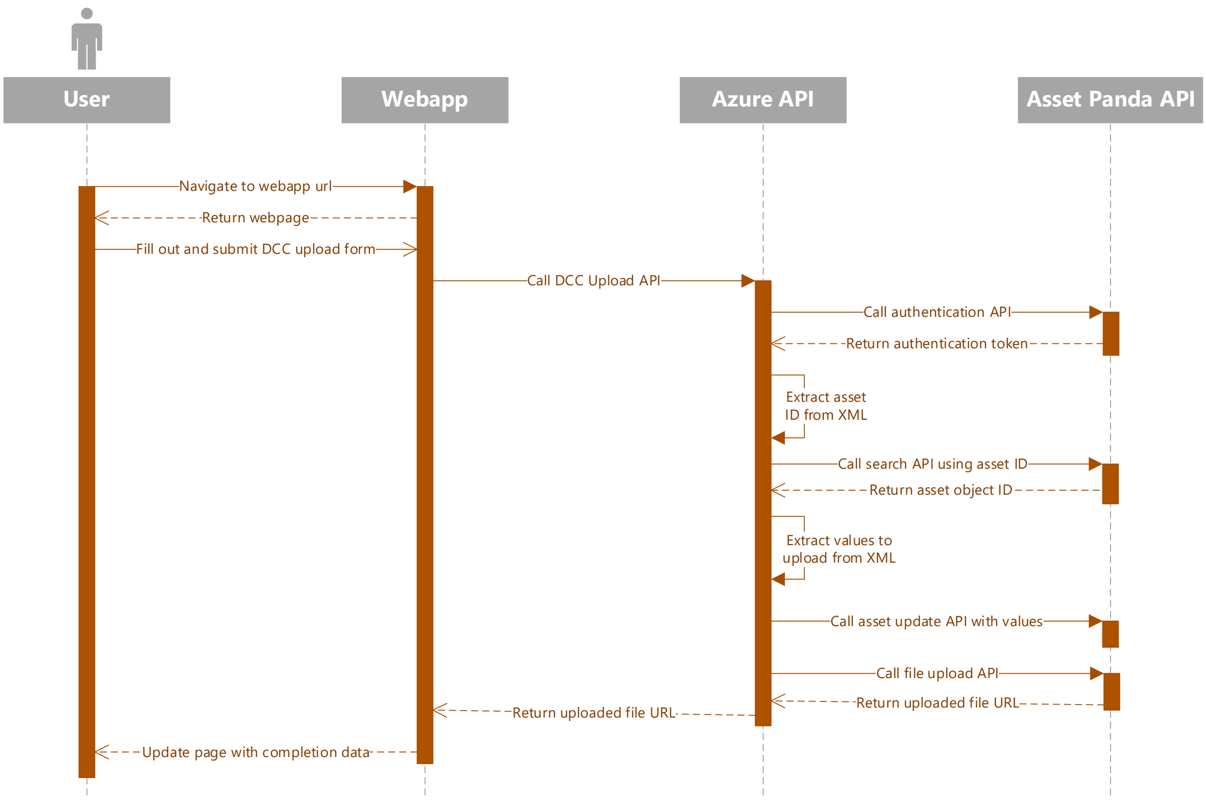

Figure 9 Sequence diagram showing how an Asset Panda record is updated when a DCC is uploaded via the webapp (see Figure 7)

This proof-of-concept demonstrated how, by exploiting the standardised, digital, XML format of the DCC, a largely automated workflow can be implemented to manage calibration records using readily available software tools. In doing so, the availability and accuracy of calibration data and records can be improved, whilst reducing the reliance on a human to upload and update information.

The exchange of DCCs between calibration service provider and customer was not addressed as part of this work since the mechanism and infrastructure to enable secure, authenticated DCC exchange has yet to be confirmed.

5.2 DCC EXPLOITATION

The following use case was proposed as an exemplar for how the use of DCCs could benefit engineering activities. The use case relates to the Ultra High Rate Deposition Cell (UHRDC) at NCC, pictured in Figure 10, which is a manufacturing capability designed for the deposition, forming and trimming of dry fibre preforms used in the production of large composite structures. The line comprises a gantry-based automated system which enables the control of 5 end-effectors, one of which carries an ultrasonic knife (USK) for near-net-shape trimming of preforms.

Figure 10 Ultra High Rate Deposition Cell at NCC

The positional accuracy of a cut performed by the USK end-effector is dependent on the position of the USK blade relative to the tool on which the preform is trimmed. This, in turn, is dependent on the accuracy of the end-effector tool centre point (TCP) information that is used by the control system.

Currently, TCP information for end-effectors is hard coded into the control system and can only be revised manually by a suitably qualified person. This means that updates to the TCP information following, for example, a change of blade tip or a collision of the end-effector are not routinely performed, which may contribute to cutting inaccuracies. Records of historical TCP information may also be unreliable. The use case explores how DCCs for the USK end-effector could be generated and used to implement a robust, traceable workflow for TCP updates.

The USK end-effector can be considered a calibrated artefact, whereby the translation between the gantry TCP and the end-effector TCP is the value to be calibrated. As such, an entry for the end-effector was created on Asset Panda, see Figure 11, to host and manage corresponding calibration records.

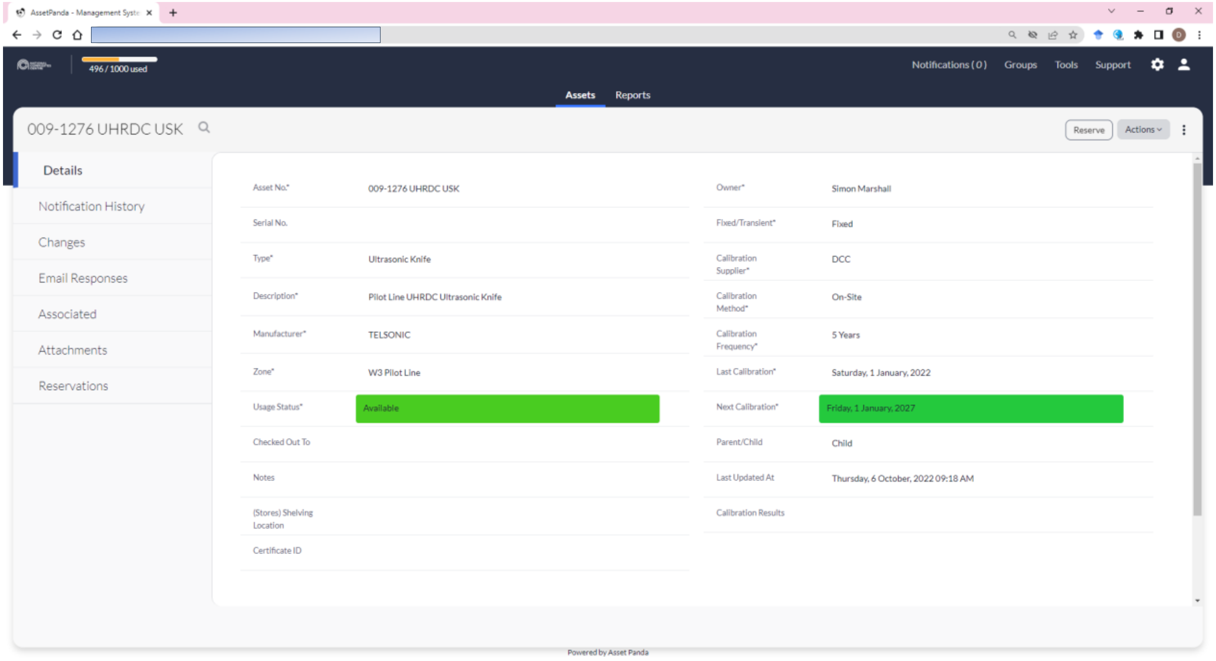

Figure 11 Asset Panda calibration record for the USK end-effector

Next, a calibration procedure was developed, during which a laser tracker could be used to measure the position of the end-effector TCP relative to the enhanced referencing system (ERS) of the manufacturing cell, as shown in Figure 12. The laser tracker used to perform the calibration procedure is itself calibrated annually, thus providing traceability back to the definition of the SI unit.

A summary of the procedure is as follows:

- Laser tracker is positioned at a suitable location within the manufacturing cell and booked into the cell ERS.
- USK end-effector is fitted with a pin-nest to hold a spherically mounted retroreflector (SMR) in place of a knife blade.
- Cell gantry collects the USK end-effector.
- End-effector is driven to a pre-defined TCP location in the cell.
- End-effector is driven to 9 orientations around the TCP; at each orientation, measurements of the actual TCP (as represented by the nest-mounted SMR) are taken using the laser tracker.
- TCP orientation measurements are repeated at a number of other pre-defined locations within the cell working volume.
- Measurement results are analysed to establish the error between the programmed and measured TCP values at each location.
- Where possible a mean error is established across the measurement locations.
- TCP positional error (or mean thereof) is reported.

Figure 12 USK end-effector TCP was measured using a laser tracker

The inspection and analysis software used at NCC, InnovMetric PolyWorks Inspector, enables the automated generation of a customised inspection report. This enabled TCP positional error, as measured via the calibration procedure, to be reported in the format of a DCC, which could be uploaded to the USK end-effector entry on Asset Panda (section 5.1).

Subsequently, calibration results for the end-effector were programmatically recalled from Asset Panda using a custom Python script. Depending on the magnitude of the reported error, an updated copy of the control system file containing the relevant TCP information could be generated, containing the latest, as-measured information from the DCC.

Replacement of the control system file itself, to enable automated TCP revision based on end-effector DCC data, was not feasible during this work due to certain constraints of the control system architecture. Nonetheless, the case study demonstrated how DCCs could be used to establish a highly automated, traceable workflow by which a manufacturing system could be updated to reflect the latest physical state of the system, thus reducing the likelihood and/or severity of positional deviations occurring during the manufacturing process.

6 CONCLUSIONS

In the next few years, many calibration services will be updated to generate and deliver machine-readable calibration certificates, referred to as “digital calibration certificates” or simply “DCCs”, rather than the paper or electronic certificates that are currently delivered.

For calibration service providers such as NPL, there are a number of challenges to be addressed. For example:

- Understanding how information in current calibration certificates can best be presented within the DCC framework.
- DCC checking, signature and security.
- Integration of DCC generation into existing calibration services.
- Development of an infrastructure to facilitate DCC delivery.

Calibration service users such as NCC also face challenges. For example:

- Development of an infrastructure to manage DCCs.
- Understanding the structure and content of DCCs.
- Development of software to automatically extract information from DCCs so that it can be used elsewhere.

It is not the intention of calibration service providers to switch from providing current calibration certificates to DCCs “overnight” and to leave calibration service users to “fend for themselves”. Instead, there will be engagement with calibration service users to:

- Raise awareness of the intention to move to DCCs, highlighting the benefits they offer (reduction in errors, fewer bottlenecks in processes, etc.).
- Provide training material to aid understanding of what DCCs are, their structure, etc.
- Provide guidance and example software on how information within DCCs can be extracted.

The case study presented in this report represents first steps in addressing some of the challenges listed above. Although the focus of the case study was purely on the information, i.e., administrative data and measurement results, within DCCs, for both participants it has proven very useful:

- For NPL, it provided an opportunity to work more closely with real world DCCs, and more specifically to identify how information intended to populate a DCC might best be stored, to consider the development of software for DCC generation, with emphasis on its reusability within calibration services, and to begin to consider how current calibration services can be updated to generate and deliver DCCs.
- For NCC, it enabled an exploration of the current state of the DCC framework and its readiness for adoption into industry, whilst also providing an opportunity to generate tangible use cases to illustrate the potential benefits of DCCs within a manufacturing environment.

ACKNOWLEDGEMENTS

NPL activity was supported by the UK Government's Department for Science, Innovation and Technology (DSIT) through the UK's National Measurement System (NMS) programmes.

NCC activity was supported by the Digital Engineering Technology & Innovation (DETI) programme, a strategic programme of the West of England Combined Authority (WECA), delivered by the National Composites Centre (NCC) in partnership with the Centre for Modelling & Simulation (CFMS), Digital Catapult, the University of the West of England (UWE), the University of Bristol, and the University of Bath.

REFERENCES

[1] Website for the EMPIR project SmartCom. ptb.de/empir2018/smartcom/project/.

[2] PTB web area on digital calibration certificates. ptb.de/dcc.

[3] Website of the National Physical Laboratory. npl.co.uk.

[4] Website of the National Composites Centre. nccuk.com.

[5] Extensible Markup Language (XML). w3.org/XML/.

[6] Introducing JSON. json.org/json-en.html.

[7] Website of Asset Panda. assetpanda.com.

APPENDIX A TEMPLATE XML DCC

The template XML DCC is reproduced below. It contains dummy textual element contents (shaded yellow).

```
<?xml version="1.0" encoding="UTF-8"?>
<dcc:digitalCalibrationCertificate xmlns:dcc="https://ptb.de/dcc"
                                   xmlns:si="https://ptb.de/si"
                                   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                   schemaVersion="3.1.2"
                                   xsi:schemaLocation="https://ptb.de/dcc dcc.xsd">
  <dcc:administrativeData>
    <dcc:dccSoftware>
      <dcc:software>
        <dcc:name>
          <dcc:content>Software name</dcc:content>
        </dcc:name>
        <dcc:release>Software release</dcc:release>
      </dcc:software>
    </dcc:dccSoftware>
    <dcc:coreData>
      <dcc:countryCodeISO3166_1>GB</dcc:countryCodeISO3166_1>
      <dcc:usedLangCodeISO639_1>en</dcc:usedLangCodeISO639_1>
      <dcc:mandatoryLangCodeISO639_1>en</dcc:mandatoryLangCodeISO639_1>
      <dcc:uniqueIdentifier>Unique identifier</dcc:uniqueIdentifier>
      <dcc:beginPerformanceDate>1111-11-11</dcc:beginPerformanceDate>
      <dcc:endPerformanceDate>1111-11-11</dcc:endPerformanceDate>
      <dcc:performanceLocation>laboratory</dcc:performanceLocation>
    </dcc:coreData>
    <dcc:items>
      <dcc:item>
        <dcc:name>
          <dcc:content>Item name</dcc:content>
        </dcc:name>
        <dcc:manufacturer>
          <dcc:name>
            <dcc:content>Manufacturer name</dcc:content>
          </dcc:name>
        </dcc:manufacturer>
        <dcc:identifications>
          <dcc:identification>
            <dcc:issuer>manufacturer</dcc:issuer>
            <dcc:value>Identification value</dcc:value>
          </dcc:identification>
        </dcc:identifications>
      </dcc:item>
    </dcc:items>
    <dcc:calibrationLaboratory>
      <dcc:contact>
        <dcc:name>
          <dcc:content>Contact name</dcc:content>
        </dcc:name>
        <dcc:eMail>Contact email address</dcc:eMail>
        <dcc:location>
          <dcc:city>Contact city</dcc:city>
        </dcc:location>
      </dcc:contact>
    </dcc:calibrationLaboratory>
    <dcc:respPersons>
      <dcc:respPerson>
        <dcc:person>
          <dcc:name>
            <dcc:content>Responsible person name</dcc:content>
          </dcc:name>
        </dcc:person>
      </dcc:respPerson>
    </dcc:respPersons>
    <dcc:customer>
      <dcc:name>
        <dcc:content>Customer name</dcc:content>
      </dcc:name>
      <dcc:eMail>Customer email address</dcc:eMail>
```

```
      <dcc:location>
        <dcc:city>Customer city</dcc:city>
      </dcc:location>
    </dcc:customer>
  </dcc:administrativeData>
  <dcc:measurementResults>
    <dcc:measurementResult>
      <dcc:name>
        <dcc:content>Measurement result name</dcc:content>
      </dcc:name>
      <dcc:results>
        <dcc:result>
          <dcc:name>
            <dcc:content>Result name</dcc:content>
          </dcc:name>
          <dcc:data>
            <dcc:quantity>
              <dcc:name>
                <dcc:content>Quantity name</dcc:content>
              </dcc:name>
              <si:real>
                <si:value>1.0</si:value>
                <si:unit>Quantity unit</si:unit>
              </si:real>
            </dcc:quantity>
          </dcc:data>
        </dcc:result>
      </dcc:results>
    </dcc:measurementResult>
  </dcc:measurementResults>
</dcc:digitalCalibrationCertificate>
```

The template XML DCC contains minimal information in the sense that the minimum number of textual element contents required for an XML DCC to be valid according to the D-SI and DCC schemas is twenty-five.

Note that it is necessary to provide at least one piece of information within the element `dcc:location`. In the template XML DCC, within both the elements `dcc:calibrationLaboratory` and `dcc:customer`, a single piece of information is provided using the element `dcc:city`. Any other child element of `dcc:location` element could equally be used.

APPENDIX B XML ELEMENTS, XMLSCHEMA AND PYTHON DICTIONARIES

B.1 ENCODING XML TO DICTIONARIES

In Python, a dictionary is a means of storing data as two components – a key and a value. The Python library `xmlschema` accounts for the structure of an XML file as defined in its schema and encodes XML elements as dictionaries as follows.

Example 1 – Element may occur once only
The element

```
<name>string</name>
```

is encoded to the dictionary

```
{'name': 'string'}
```

where `'name'` is the key and `'string'` is the value.

Example 2 – Element may occur more than once
Two instances of the same element, such as

```
<name>string1</name>
<name>string2</name>
```

are encoded, in a compact manner, to the dictionary

```
{'name': ['string1', 'string2']}
```

using a list of strings.

Example 3 – Parent element may occur once only, child element may occur more than once
The element

```
<name1>
  <name2>string</name2>
</name1>
```

is encoded to the nested dictionary, i.e., dictionary within a dictionary,

```
{'name1': {'name2': ['string']}}
```

Even though the child element occurs only once, the fact that the schema allows it to appear more than once results in `xmlschema` using a list of one string for the child element.

B.2 DECODING DICTIONARIES TO XML

When using `xmlschema` to decode a dictionary to XML, there is flexibility in how the dictionary may be provided. Examples 1 and 3 of section B.1 are revisited as follows.

Example 1 – Element may occur once only

The dictionaries

```
{'name': 'string'}
```

and

```
{'name': ['string']}
```

are both decoded to the element

```
<name>string</name>
```

Example 3 – Parent element may occur once only, child element may occur more than once

The dictionaries

```
{'name1': {'name2': 'string'}}
```

and

```
{'name1': {'name2': ['string']}}
```

and

```
{'name1': [{'name2': 'string'}]}
```

and

```
{'name1': [{'name2': ['string']}]}
```

are all decoded to the element

```
<name1>
  <name2>string</name2>
</name1>
```

In the Python software developed for the case study:

- A list is not used for an element that is permitted by the DCC schema to occur only once.
- A list is always used for an element that is permitted by the DCC schema to occur more than once, even if the element occurs only once.

The dictionaries shaded yellow in the examples above represent how the element would be encoded by the Python software.

Adopting a consistent approach helps to reduce the complexity of code as there is no need to count the number of instances of an element that may occur more than once and then decide whether or not to use a list based on that number. In addition, the approach facilitates code reuse.