United States Air Force Metrology and Calibration Program's NextGen Calibration Automation system

Speaker/Author: Marc L. Monnin
AFMETCAL DET1/MLES, USAF
813 Irving Wick Road
Heath Ohio USA, 43056
Email: Marc.Monnin@AFMETCAL.af.mil
Commercial Phone: 740-788-5135

Abstract

Automation is a balancing act between people, time and equipment. People have limited time. Computer languages and operating systems change which requiring extensive rework of existing software. Calibration hardware uses an ever-expanding plethora of various control languages and communication protocols.

The Air Force Metrology and Calibration Program (AFMETCAL)'s NextGen Calibration Automation system is one answer to the above issues. The NextGen Calibration Automation uses three defined layers that can be modified independently of one another interacting via common interfaces. The Sequencer Layer handles equipment configuration, graphical user interfaces and abstracts test logic into a readable XML standard. The Measurement Module Layer is an equipment independent layer which handles technique, timing and the taking of a measurement. Finally, the Hardware Abstraction Layer concerns itself with instrument control, communication buses and equipment substitution.

Introduction

Effective automation of calibration is a balancing act between people, time and equipment. People with extensive calibration knowledge are in high demand and have limited time for an automation project. Computer languages and operating systems change which requires extensive rework of existing software that leaves little time for new development. Calibration hardware uses an ever expanding plethora of various control languages and communication protocols. Industry tries to help with standards but they are often too little, too late or in a constant state of flux. Additionally, changes in the Air Force methodology in funding standards means that the Air Force Calibration labs will likely have a more diversified inventory of calibration standards. The Air Force Metrology and Calibration Program's internally-developed NextGen Calibration Automation system is one of AFMETCAL's attempts to address the above issues.

The NextGen Calibration Automation system uses three defined layers, the Sequencer Layer, the Measurement Module Layer, and the Hardware Abstraction Layer. These layers can be modified independently of one another. The layers interact via common interfaces. Therefore a programmer can program to the interface without the need to know the internal details of the layers. This allows the details of the layers to change independently without having to rewrite the entire program. The Sequencer Layer handles equipment configurations, the graphical user interface and test logic. The Measurement Module Layer is an equipment independent layer which handles technique, timing and makes a particular measurement. In fact, it captures calibration knowledge. Finally, the Hardware Abstraction Layer concerns itself with instrument control, communication buses and equipment substitution. The Air Force Metrology and Calibration Program has already fielded several automated oscilloscope calibration programs using the defined layers of these NextGen Protocols. This paper describes the challenges in and the reasons for using these defined layers.

Why we need NextGen

After many automation projects, AFMETCAL recognized the problems we face in automation. An abbreviate list follows

Problems Faced in Automation

1. Lack of people
2. Lack of Expert knowledge when you need it
3. Lack of time
4. Rewrites every time hardware or OS changes
5. Programmers are forced to become Metrology area experts
6. Calibration standards keep changing

So people, time and hardware changes are the major hurdles to automation.
After weeks of deliberations we came up with an obvious observation.

Obvious Observation #1: All people have different skills, experience levels and abilities.

Since people have different levels of experience and skill sets, NextGen breaks the task into parts that can be matched to skill sets to solve the problem, making sure the responsibility flows to the right people at the right time. Figure 1 demonstrates the process.

Flow	(Responsibility)
1. CRD	(Technical Content Mgr)
2. Cal parameter value	(Technical Order Writer)
3. Measurement Module	(Technical Content Mgr)
4. MM in a HLL	(Automation Team w TCM)
5. HAL	(Automation Team)
6. Instrument/Drivers	(Automation Team)

542D COMBAT SUSTAINMENT WING

Figure 1

1. The Air Force bases Calibration Procedures, also called Technical Orders (TO), on Calibration Requirement Documents (CRDs). These documents state which parameters of an instrument must be calibrated and suggest techniques to be used in the calibration. CRDs contains the science behind the calibration and provides a common vocabulary to discuss an instrument's calibration needs. CRDs are written and maintained by the measurement area calibration expert referred to as the Technical Content Manager (TCM).
2. For a particular instrument, CRDs are tailored to address the specific test points and calibration techniques needed to calibrate that instrument in the form of a TO. In the NextGen model, it is the responsibility of the TO writer to pick the specific test points and techniques and to fill in the parameters based on the manufacturer's specifications for the instrument.
3. The heart of the NextGen system is the Measurement Module (MM). The MM is a specific technique that returns a value or series of values that makes up a measurement. It is designed by a TCM who has knowledge of the limitations and expectations of its use. The MM can be thought of as a subset of the CRD.
4. The next part of the flow is to actually turn the measurement module into code. This will require that the TCM work alongside the automation programmer. The programmer uses state machines and a high level language to do this.
5. The high level language then talks to the Hardware Abstraction Layer (HAL). Because we have abstracted out the hardware, many different physical pieces of hardware can be used to perform the same function – in other words, this allows substitution of equipment without changes to the MM. This allows reuse of validated calibration procedures independently of equipment used.

6. Finally, the hardware itself talks to the communication bus via its driver. This is the layer containing the familiar SCPI and ASCII commands.

This flow (see Figure 1) puts the domain knowledge expert in the right place at the right time. It is respectful of people's time and maximizes what their expertise. The TCM knows the technique and science of a measurement and is therefore in the unique position to design a MM. Once a MM is validated, it is available to be used in any calibration increasing reuse and saving time. Programmers don't have to become minor TCMs to do their jobs. They can now focus on what they do best and, by using proper software abstractions, they can alleviate the need to completely rewrite a system every time an operating system or computer language changes. Finally, TO writers are the most familiar with the specifications and requirements of a particular instrument. Letting them put in the specific test points ensures that the calibration parameters meet USAF specifications.

How we do NextGen

I have talked about who does what. Now, I would like to discuss the specifics of how it is done. As previously mentioned, NextGen Calibration Automation uses defined layers that can be modified independently of one another, but interact via common interfaces.

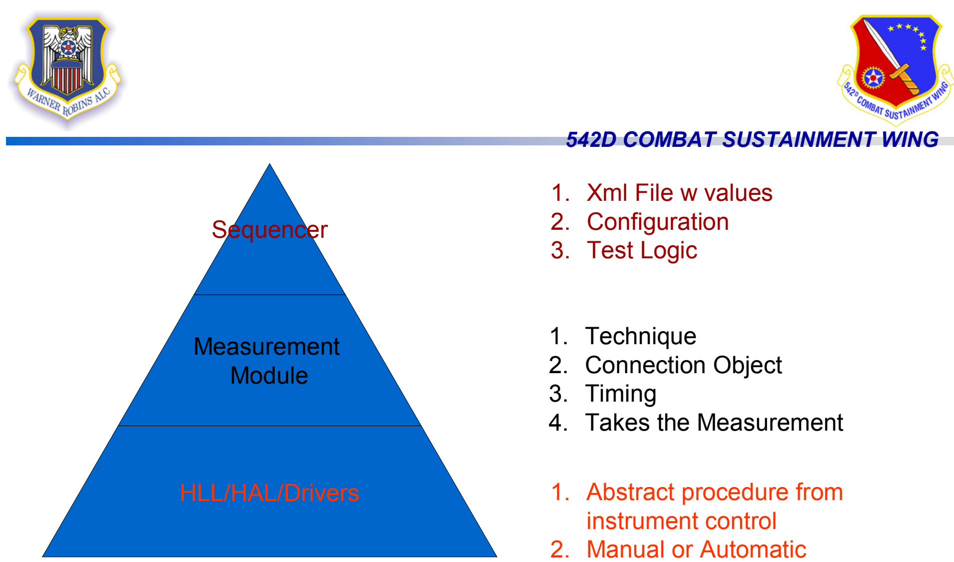

Figure 2

Figure 2 shows these layers as a pyramid, and indeed they are. Each one builds on the previous layer. The layers are independent and only know of each other through their public interfaces. It is a NextGen system goal to be able to switch out and replace the different layers without affecting the others. This will allow the system to change and grow without total rewrite of the code base.

The Sequencer Layer handles equipment configuration, graphical user interfaces and test logic. It is responsible for running the control scripts that the TO writer builds for each specific instrument. These scripts are in a human readable XML text file. A key feature is that the script is written in Extensible Markup Language (XML). XML is a markup language for documents containing structured information. In other words, it contains metadata, that is data that describes the data it contains. While it is not completely transparent, the TO intention can always be deduced by looking at the script.

Because it is a simple text file, it can be easily transported and modified and it will never become unreadable as technology changes. An example is in order. The following is a simple DC voltage calibration for an oscilloscope where an oscilloscope calibrator applies a DC voltage with offset and the oscilloscope returns the offset that it measures. Two calibration points are included.

```
 <Test Name="DCVolts"> CRD="DigitalScopeCRD" CRDTest="DCVolts">
<Preset Method="Preset">
   <TI_TimeDiv>0.002</TI_TimeDiv>
   <Impedance>Imp.Hi</Impedance>
   <TI_NumberOfAverages>100</TI_NumberOfAverages>
</Preset>
```

Preset

```
<Table>
   <TableHeader>
      <Col Content="TI_VoltDiv" Label="Volts/Div"></Col>
      <Col Content="TI_OffsetVoltage" Label="Offset(V)"> </Col>
      <Col Content="CalibratorAppliedVoltage" Label="Applied(VDC)">
      <Col Content="LL" Label="Lower Limit (VDC)">   </Col>
      <Col Content="UL" Label="Upper Limit (VDC)">    </Col>
      <Col Content="Result" Label="Result (VDC)">  </Col>
      <Col Content="Pass" Label="Pass/Fail">    </Col>
```

Table Header

```
   </TableHeader>
   <ActionScript>
      <Action LocalObject="Result" Method=
"Measure(TI_VoltDiv,TI_OffsetVoltage,CalibratorAppliedVoltage)">
      </Action>
      <Action ColumnVar="Result" Object="Result.Result">
      </Action>
      <Action ValidationTestValue="Result">
      </Action>
      <Action ColumnVar="Pass" Validation=
   "InBand(LL,UL,ValidationTestValue)">
      </Action>
```

Action Script

```
   </ActionScript>
   <TableBody>
```

```xml
<Row>
   <TI_VoltDiv>0.002</TI_VoltDiv>
   <TI_OffsetVoltage>-0.4000</TI_OffsetVoltage>
   <CalibratorAppliedVoltage>0.4000</CalibratorAppliedVoltage>
   <LL>0.394680</LL>
   <UL>0.405320</UL>
   <Result />
   <Pass />
</Row>
```

Table Row

```xml
<Row>
   <TI_VoltDiv>0.002</TI_VoltDiv>
   <TI_OffsetVoltage>0.4000</TI_OffsetVoltage>
   <CalibratorAppliedVoltage>-0.4000</CalibratorAppliedVoltage>
   <LL>-0.405320</LL>
   <UL>-0.394680</UL>
</Row>
```

Table Row

. . . .

The above example shows the benefits of using XML. It is easily understood, very transportable and the metadata (data that describes the data) can be understood without an advanced degree in computer science. At AFTMECAL we go one step further and give the writers of the file a tool that hides even the metadata. All that the writer sees are tables and sections containing the applied values and validation limits for the points they want to test. Using this tool they can quickly write their calibration which, because of the tool's data validation, is guaranteed to run.

The Sequencer layer also builds the configuration to set up specific instruments and the connections that the Measurement Modules need to run. It is also in charge of the engine that interprets the XML files and performs the validation of the Measurement Module results.

The next layer brings us to the heart of the NextGen system, the Measurement Module. All the measurement science and techniques are incorporated into the Measurement Module. The Measurement Module's purpose is to take a measurement. The Measurement Module knows how it is connected to the instrument under calibration. It knows its connector type and any connector hook up that will occur. It knows how accurate the measurement it makes is based on the instruments and cables that are used. It knows how many and the type of assets it controls. It handles timing of both source instruments as well as the measurement process. It also knows how many waveforms to count, how long to wait for a result and what to measure.

An example should clarify what the Measurement Module is. Assume you want to measure AC volts on an oscilloscope. There are many ways to do this. A simple scope calibrator could be used or perhaps a TVC and an uncalibrated AC source could be used (figure 3). Both of these would become different measurement modules because, even though they are measuring the same parameter (Voltage), they use different techniques and connections and assets. The Measurement Module would apply a known voltage through the calibrator; allow the voltage to settle, set up the oscilloscope to measure the signal and measure the average voltage after number of cycles pass. The number of cycles, frequency and applied voltage would be parameters passed into the Measurement Module. The TCM would determine how the voltage was measured and how accurate a reading needs to be. This would tell the TO writer when to use this measurement module. Since this module is tied into a CRD there would only be a few AC volts techniques for the TO writer to concern themselves with. And since a Measurement Module contains a full description of its technique and uncertainties, the TO writer can quickly pick the appropriate module from a list of valid modules.

In the above example, the AC voltage source could be a meter calibrator or an oscilloscope calibrator, providing that the instrument meets the requirements for uncertainty, frequency and amplitude. The Measurement Module only knows that it has a voltage source that hooks up to the instrument under calibration. The module does not care what voltage source is used as long as it meets the necessary specifications.

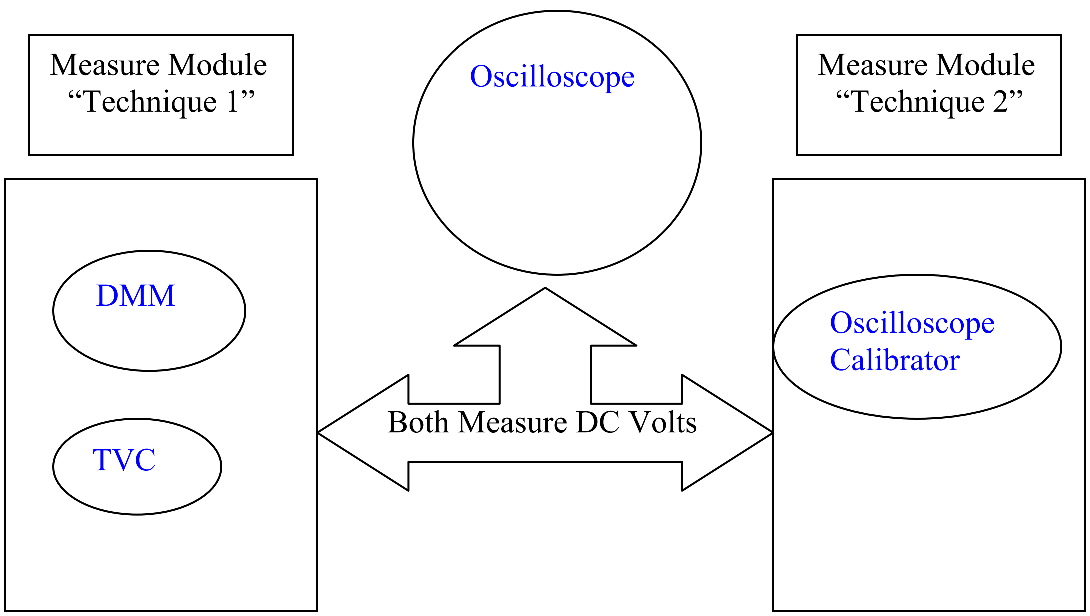

Figure 3

Measurement Module techniques are developed by the TCM. The automation team codes it, but the TCM verifies that it does the job and that it meets the required specifications. If one calibration lab does not have on type of source, it can switch in a different asset that meets or exceeds the requirement. Thus, calibration labs can use any acceptable standard that meets the MM requirements without the need to change the software code for the MM. This ability to switch a standard in a Measurement Module gives great flexibility.

The last layer is the Hardware Abstraction layer. The purpose of this layer is to abstract communication from vendor specific details for a particular item. This includes instrument specific drivers and bus specific protocols. Coding all communication to an interface that called the HLL (high level language) does this. Since all communication in the layer above (the Measurement Module) is written to this interface standard, the specific communication code can be changed without affecting the rest of the system. This allows us great freedom. We only need to code a translator from the HLL to a specific bus communication layer.

Finally, each physical instrument command needs to be written in a vendor specific language. We abstract out the instrument into independent drivers and have each driver call a base instrument class for the specific instrument type, such as a base meter calibrator. The base instrument class allows us to code independent of the specific vendor and to test the base layer without the need of an instrument. The specific drivers are independent of the hardware bus communication. This allows us to use the same driver on GPIB-488 as easily as IP or USB.

Conclusion/Summary

Abstraction to the different layers allows us great freedom to switch and modify parts of the NextGen system without needing to rewrite the whole system. Because of the limited number of programmers available extensive testing is used. Each layer and each part of each layer are tested independently. There is as much test code as production code. These tests give us great confidence in our ability to modify the whole system or any part of the system without introducing computer bugs.

We run our tests early and often. If one of the programmers introduces a change that affects another's existing code, the tests will quickly point to the culprit code. The two programmers can then come to agreement on how to proceed.

NextGen Strengths

1. We work as a team.

2. Each member does what they are uniquely qualified to do (programmers program, TCM do the science, TO writer's put it altogether)

3. Abstract the layer to minimize rework.

4. Design measurement modules to take one measurement using a specific technique

5. Design to Interfaces and abstract out what changes

6. Test a lot.

The AFMETCAL NextGen Calibration Automation system is a system of people, science, technology and tools. It leverages people's abilities and knowledge. It captures calibration science in reusable modules, technology and tools to make it all happen.