Evolving System Integration Labs in the Era of Digital Transformation: A Brief Primer on Digital Transformation and its Implications for Test Systems and SILs

William N. (Bill) Eccles
Principal Engineer, Director of Corporate Development
Bloomy
South Windsor, CT, USA

ABSTRACT

The nature of systems and product engineering is changing rapidly with the introduction and application of Digital Transformation and a toolset consisting of Model-Based Systems Engineering, the Digital Thread, and Digital Twin. Each of these tools is briefly introduced and examined, and several characteristics for test systems which are necessary for full digital transformation implementation are derived and enumerated. The progress that the author's test equipment manufacturing and design firm is making towards providing COTS products which interact with models, the digital thread, and the digital twin is reviewed. Finally, a brief case study is reviewed which shows a 30% increase in test efficiency after implementing one of the enumerated characteristics, which was made possible through the use of COTS products.

INTRODUCTION

Bloomy has been applying automated test technologies into all areas of the product lifecycle, from R&D and validation to production and MRO, for the last 30 years across commercial and military applications. Our position as external resources for numerous organizations provides a unique perspective on the state of test systems and allows us to develop systems and methods which address multiple industries' needs. After consistent observation of the mil/aero industrial complex's struggles with adopting COTS technologies or with maintaining custom software and hardware in-the-loop systems, we began an initiative starting in 2015 to infuse the HIL market with COTS solutions which address these struggles head on. These COTS solutions have enabled enhanced test efficiency and reduced system delivery times, and were key to reducing reliance on overburdened internal technical resources.

During this same period, "Digital Transformation" rose to buzzword-worthy status to the point where it was largely an over-applied term and lost its meaning. [Ref. 1] However, the systems engineering profession took Digital Transformation very seriously and has made great strides in rearchitecting systems engineering to accommodate increasingly-complex systems. The International Council on Systems Engineering (INCOSE) introduced concepts, such as model-based systems engineering, the digital thread and twin, which embody the intent of Digital Transformation. Because our customers' products are being used by organizations which are in various stages of adopting these transformational methods, we are actively adapting our systems to fully participate in a Digitally Transformed systems engineering environment.

It is reasonable to assume that accommodating Digital Transformation in test systems means merely building them in a more flexible way in order to keep up with the ever-increasing pace of the product cycle and the risks of last-minute changes—especially during validation testing which usually occurs coincident with product development. Though this assumption is true, it is only partially responsible for the necessity for test systems to be fully Digital Transformation-capable, and it certainly does not address test systems outside of the validation phase. As a provider of test systems which address all phases of a product's lifecycle, Bloomy recognizes the imperative for all test systems, from concept to sustainment, to be more than just flexible and responsive, but also to be interactive with the new initiatives of Digital Transformation-driven changes coming to systems engineering.

As Bloomy has seen in our customer base, "digital transformation" is being implemented in numerous different ways and at inconsistent rates. Based on our observations, we will describe at a fairly high level what we believe Digital Transformation means to our customers by describing INCOSE's vision of digital-transformation-enabled systems engineering. We'll then show why we believe test systems of all kinds must be able to interact with the tools and methods of the

Presented at the VFS Modern Ops. with Future Integrated VMS Technical Meeting, West Chester, PA, Aug. 19-20, 2021.

vision. We'll then show how Bloomy-developed COTS technologies address these transformational requirement. Finally, we'll describe a brief, simple example of early, but significant, improvements obtained using these technologies in a systems integration lab (SIL) for helicopter flight controls.

SYSTEM COMPLEXITY

In the beginning, systems engineering was done with paper and pencil, draftsmen and women, and with real-world system testing—risky tests placing humans at risk, especially in vehicular systems. As available technologies increased, so did system complexity. Figure 1, a graph of "thousands of lines of code" for various aircraft models, gives us one representation of just how complexity has grown in one span of 30 years.

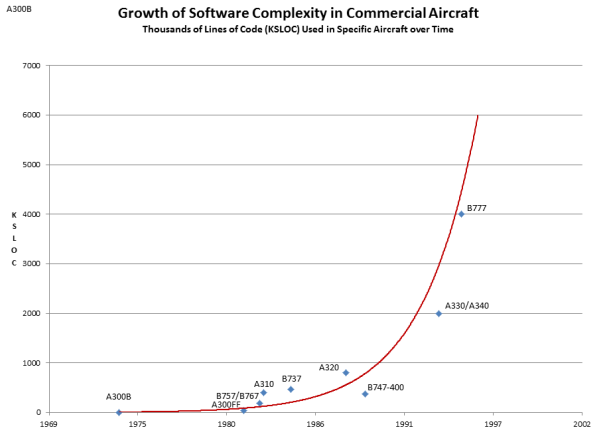

Figure 1: Thousands of Lines of Code (KSLOC) Used in Specific Aircraft over Time [Ref. 2]

One noted scholar, Christopher Alexander, has recognized that systems are too hard to predict and test.

> "Today more and more design problems are reaching insoluble levels of complexity."
>
> "At the same time that problems increase in quantity, complexity and difficulty, they also change faster than before."
>
> "Trial-and-error design is an admirable method. But it is just real-world trial and error which we are trying to replace by a symbolic method. Because trial and error is too expensive and too slow." [Ref. 3]

Given the graph of Figure 1, it would be easy to think that Alexander recognized this complexity in the mid-Nineties. However, he wrote these words in *1964,* well before software was ever found on aircraft. So it's clear that complexity was a recognized problem well before systems became as complex as they are today.

In the intervening years, of course, we've advanced to the leading edge of Industry 4.0, the age of cyber-physical systems. These cyber-physical systems consist of interconnected processes and products, and the number of possible interconnections between them has risen to the point of uncountability. A "simple" network of twenty nodes, where a node can be a physical device or a software process on a device, can be configured in 2^{190} different ways. [Ref. 4] Clearly, we can't test each of these configurations.

The number of interactions that we need to test, the configurations we need to plan for, and the nature of the systems we can build in the connected age of the "internet of things" is so large that it has eclipsed our ability to manage it. We need some method to ensure that the complexity is managed to a reasonable level, or we need something that can help us test all of the "important" interactions.

Taming Complexity with Model-Based Systems Engineering

Model-Based Systems Engineering (MBSE) can help manage the increase in system complexity. [Ref. 5]

Formally, INCOSE defines MBSE as:

> "…the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases." [Ref. 6]

MBSE is significantly more complex than what is presented here, [Ref. 7] but in practice, MBSE has several major characteristics.

First, it is a replacement for document-based systems engineering (DBSE), where many different distributed documents contain the characteristics, implementation, and requirements of the system. Though we've advanced from paper copies of documents to electronic copies of documents, they're still just as easy to lose and difficult to find amongst the various silos in the typical large project group. Furthermore, they are poorly revision controlled. Products such as IBM's Rational DOORS have made inroads into configuration control and traceability of requirements, tests, implementation, etc., but the data in these sorts of tools remains inconsistently expressed, making machine usage of the data an impractical task of exporting and processing before it can be used for automated purposes—a process that has to be repeated each time the data are updated.

These limitations give rise to the second and third major characteristics of MBSE: the model is expressed in a formal, machine-parsable language, System Markup Language (SysML), no matter what kind of data (whether financial, engineering, performance, etc.) is stored; and it is configuration controlled using blockchain or equivalent technologies. The last major practical characteristic of MBSE is that it is the sole, authoritative source of truth. No other repository of the data exists, thus ending countless versions of PowerPoint presentations floating around on network drives with conflicting information.

These characteristics of MBSE are enablers for other benefits. For example, the machine-parsable nature of the model

allows for interaction with the model's data in previously difficult (or impossible) ways which can enhance our abilities to develop, update, maintain, and operate test systems. With proper expression of the system's interface types and quantities, the test system's design can be automatically generated if not completely, then at least in part, and can be updated quickly as requirements can, and will, change. If test requirements and limits are contained in the model, test scripts can automatically be generated and updated as well. From the perspective of a test engineer, the more data available in the model relevant to the test system and unit under test, the more opportunities there are to automate the life cycle and operations of the test systems.

The interaction between test systems and models isn't constrained to unidirectionality. A bi-directional relationship between them enables, for example, automated analysis of results, tuning of the model, predictions of reliability and trend analysis, randomized "fuzz" testing and coverage analysis, and, with the latest developments, [Ref. 8] artificial intelligence analysis of the system and data. And because of the nature of the model, all results are tied back to system requirements in the ultimate form of traceability.

Whether the test system is involved in verification, validation, production, or field support, for most efficient and effective development, maintenance and operation, the system must be able to interact with the model. Though some companies have made strides in this direction, the toolsets of MBSE are still largely unconnected to the test system lifecycle. Bloomy is working to fill this gap with our Digital Enablers for Test (DEFT™) which will include connectors between MBSE toolchains, our test equipment design processes and our customers' test scripts and methods.

Enabling Agility with Digital Twins and Digital Threads

Two other tools in the digital transformation toolkit are the digital thread and digital twin.

The connection between the model and the product's manufacturing processes was initially called the *digital thread*. The term originated at Lockheed Martin sometime prior to 2011 and "described using 3D CAD data to directly drive CNC (computer numerically controlled) milling or composite programming systems for carbon fiber placement. In both cases, the physical piece is the result of *an unbroken data link* that stretches back to the original computer model of the part; *the unbroken data path* was the digital thread." [Ref. 9] (author's emphasis)

As emphasized above, the key concept of the digital thread is the *unbroken* connection path that the data take from origin to destination. Similar to the chain of custody for evidence, the digital thread remains in tact, traceable, with no disconnects between source and destination. In practice, an updated requirement would flow through the digital thread and influence the manufacturing process appropriately.

Eventually, though, the idea of digital thread extended past the CAD data and CAM programs in both directions leading to the concept of a *digital twin*. As described on DigitalEngineering247.com, "In the [2013] USAF Global Horizons report on technology visions, digital twin is defined as 'the creation and use of a digital surrogate of a material system to allow dynamic, real-time assessment of the system's current and future capabilities.' The 'digital surrogate' is a 'physics-based technical description of the weapon system resulting from the generation, management and application of data, models and information from authoritative sources across the system's lifecycle.'" [Ref. 10] Here, for what is reportedly the first time, is a definition of a digital twin (or digital surrogate), and the definition makes clear that the digital twin involves models and all data associated with the operational and future operational characteristics of a system.

What is not made terribly clear is that the digital thread is the enabler of the digital twin, with the digital twin being a high-fidelity digital representation mirroring the lifespan of a *particular* product and *particular* serial number. The digital twin can be one or more high-fidelity computational models, combination of models, or tools which simulate and capture the life history of the related instantiation of the product. [Ref. 11]

In the same report, the USAF advocates use of a "tightly-integrated digital thread and prototyping process to enable *agile* development." The report also describes the USAF's view of the combination of the digital thread and digital twin as the "game-changer that provides the *agility* and *tailorability* needed for *rapid* development and deployment, while also reducing risk." (author's emphasis, both excerpts)

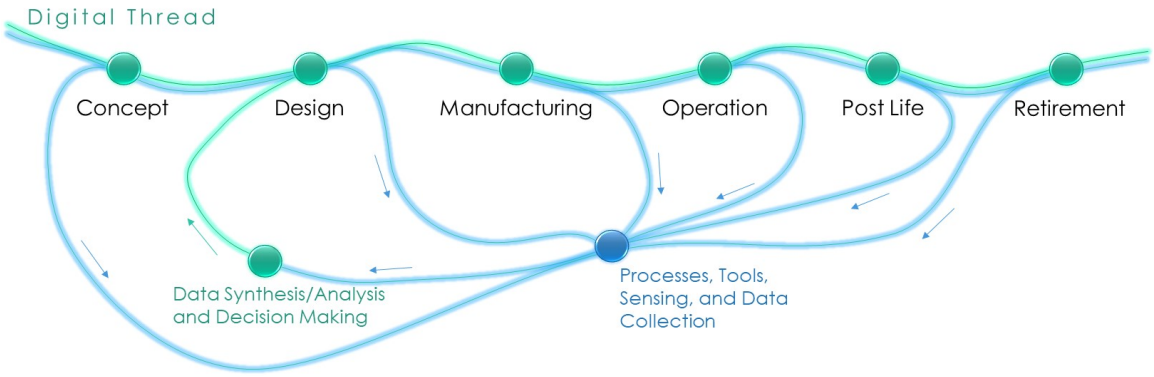

Figure 2: Illustration of engineering design with Digital Thread [Ref. 12]

Figure 2, from an academic paper which does an admirable job of expressing the mathematics of digital-thread-based design, shows an excellent representation of the way the digital thread connects both forward and backward through the product lifecycle.

Feedback from late stages can be used to influence future choices and reduce uncertainty in design parameters and process costs. This information may also reveal more efficient operational strategies, and every time these decisions are made. Each decision changes the state of the digital thread.

Taken together, the USAF's identification of digital thread/digital twin processes and the above graphic which expresses the need for the product to be able to react to both early-life changes as well as late-life discoveries, emphasizes that any test equipment used during these processes may be called upon to change as rapidly as the product may change. During early-stage iterations of the product's design, risk reduction tests using subsets of the system may be needed. But then as the product grows, so must the test system, preferably in such a way as to preserve both the investment in software and hardware made to date. This problem, which involves both hardware agility and software agility, has been solved to a large extent with Bloomy's Digital Enablers for Test (DEFT™) and the Bloomy architecture for Mil/Aero Simulation Systems, both of which are described below.

Distilled Characteristics of Digital Transformation-Ready Test Systems

In the previous sections, we've enumerated several characteristics of digital transformation which have direct implications for test systems. Distilled below are these characteristics:

1. The test system design must be able to be derived and controlled directly from the digital thread. Each design element must be traceable to an underlying requirement. Automatic, "pushbutton" test system design and manufacturing would be ideal.
2. The test system must be able to interact with the system model by retrieving tests, test limits, measurements, design parameters, methodologies, etc., from the model and digital thread. The resulting tests must be traceable to an underlying requirement.
3. Changes to the model and/or digital thread must drive changes to the test system's hardware and software. The software and hardware should be implemented in such a way that the cost of change implementation is not prohibitive.
4. Corollary: the underlying mechanisms of the hardware and software implementation should be agile and flexible in nature, easily changed quickly.
5. Data generated by test systems must be returned to the digital thread and/or digital twin and must be compatible with analysis tools which may be used to improve the system/product.
6. None of these requirements are dependent on any other, and any level of implementation of these requirements should yield an improvement to the product lifecycle.

We assume that this list is far from exhaustive, but it's at least a good start. We are also sure that this list will continue to evolve as the state of digital transformation continues to advance.

ACHIEVING DIGITAL TRANSFORMATION FOR TEST SYSTEMS: ONE COMPANY'S PROGRESS

At Bloomy, we've already tackled some of the above requirements at some level in the course of improving our own products and processes. Two products that we currently provide are used to partially address these requirements, and we're actively developing the products to further fulfil these needs. In this section, we'll describe the EFT product, which is becoming the DEFT™ product previously referenced, and the Bloomy Mil/Aero Simulation Reference System.

The EFT, originally so-named because it was designed as a software toolkit for Electronics Functional Test, provides many capabilities which Bloomy has used in our own customer deliveries for well over a decade. It grew out of our own need to meet some of the above requirements in our own

development processes, and as we wrote the software, we recognized that it had intrinsic value to customers who would otherwise not purchase a test system from us and thus sell it as an independent product.

One major function which the EFT provides is hardware and measurement abstraction layers. An *abstraction layer* is a software construct which disconnects a higher-level of software from an underlying implementation, whether hardware or software. In the real world, a steering wheel is an abstraction layer because you can turn it and the vehicle's wheels turn whether the underlying steering rack is manual or power. Similarly, abstracting away the hardware in a test environment allows tests to be written irrespective of the underlying hardware. The test may be written in a hardware-agnostic fashion, making the test more portable and making it much easier to change the underlying hardware.

Similarly, a measurement abstraction layer allows the test script to contain an instruction to take a measurement without any underlying knowledge of the hardware or software required, perhaps using a set of hardware. As Figure 3 shows, the measurement abstraction layer is a higher level of abstraction than the hardware abstraction layer. It may be a complex MAL, where the underlying hardware is completely unknown, or it may be less complex and have some knowledge of the underlying hardware.

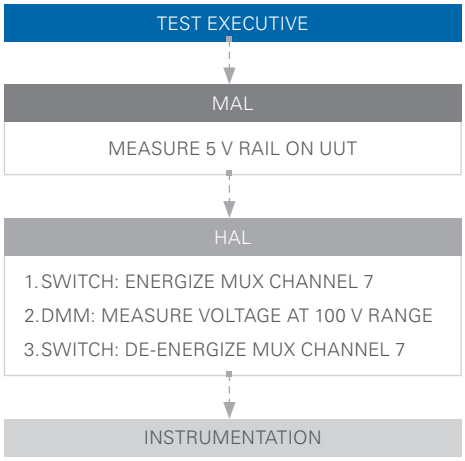

Figure 3: Abstraction Hierarchy [Ref. 13]

In this figure's example, the MAL has no idea whether a digital multimeter or an analog-to-digital converter will be used, nor does it know what switching is required to accomplish the measurement. A test script written at the HAL level would require all three steps to be explicitly written, but the same test script written at the MAL level need only know the measurement to be performed.

Both of these abstractions help attain test equipment characteristics 2-4, above. Consider that the system model in the digital thread may require "The UUT voltage must be greater than or equal to 0.5 VDC" (expressed in SysML, of course). Bloomy's MAL is able to implement the test sequences required to perform the measurement. However, a human is necessary to read the natural-language requirement and translate it into the appropriate MAL step. This is a fast process (characteristic 4), though we are working to migrate this capability to a new version of the EFT (DEFT, mentioned previously) which will be able to extract the requirement from SysML and drive the test system automatically—only one very important connection remains to be made. (Bloomy is actively seeking a lead user to work with in this capacity.) We anticipate that this step will require vendor-specific connectors for each MBSE package.

The EFT also has a plugin which addresses providing data back to a digital thread or other database (characteristic 5). It uses Structured Query Language (SQL) and a database engine connector, and for companies which use standard database systems, this capability is ready, out of the box. Again, however, where an MBSE package is being used, modification will be required to provide a vendor-specific connection to return data from the test system to the digital thread.

Similarly, we have tackled the challenges of rapid deployment of test systems by implementing a hardware design and build architecture which help address test equipment characteristics 1, 3, and 4. Twenty years ago, the state-of-the-art in test equipment system design incorporated error-prone spreadsheets, laboriously-created CAD schematics, and tedious wiring-to-print, yielding something which looked like Figure 4. Nothing was model-driven (though, to be fair, model-based design did not even exist). Changes, though possible, were certainly inconvenient and expensive. If the resources were available, several companies surveyed by the author implemented large printed circuit board assemblies which replaced the tedious wiring, but which were expensive to design and difficult to change. Again, neither design speed nor agility was a characteristic of these systems.

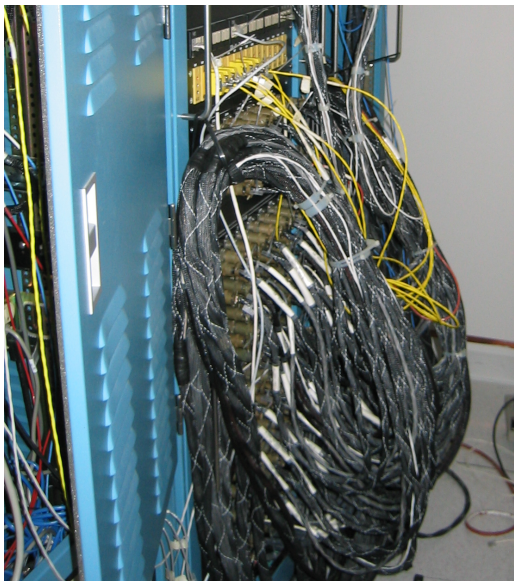

Figure 4: The back of a typical closed-loop test system circa 1999 [Ref. 14]

Since 2015, however, Bloomy has been working to develop the Bloomy Mil/Aero Simulation Reference System, which

is modular COTS hardware and brings design and implementation up from the signal/wire level to the cable level. We have also developed our Universal Test System (UTS) platform which makes use of instrumentation assemblies which can be quickly connected together to form a complete test system from a few standard assemblies. In both platforms, COTS cables are used to connect from the mass interconnect (UTS) or Bloomy ThroughPoint™ interface panel (simulation systems), and the signal conditioning and instrumentation. This implementation allows design by *cable* instead of design by *wire*, so that systems become plug-and-play and significantly reduce construction effort and troubleshooting. This build process has allowed us to reduce system design and construction times from 6-8 months to 6-10 *weeks*.

By designing and using modular systems, we can create systems from scratch more quickly and efficiently from the preliminary I/O of a prototype product. If the product's requirements change, the system can be upgraded later as more capability is needed. This flexibility is a great feature made possible by modular, COTS hardware, but it would be useless if the software configuration of this system, which is often quite complex, weren't just as flexible and easily deployable as the hardware.

Both hardware and software flexibility and agility are required to achieve all of the above requirements. We and others have a long way to go before we've achieved total interconnection with the digital thread, but we're actively working to meet this milestone by working with our customers and implementing whatever level of thread-connectedness they are ready to achieve, as we'll see next.

DIGITAL TRANSFORMATION MEETS A SYSTEM INTEGRATION LAB

In 2016, a global air, space, and defense contractor selected the Bloomy Mil/Aero Simulation Reference System for six test systems in a helicopter system integration lab. The systems were used to verify mission systems, flight controls, engine controls, and other control systems for an airframe. The technologies used which are most relevant to digital transformation are:

- NI PXI, SLSC, VeriStand, TestStand and Requirements Gateway
- IBM Rational DOORS
- Bloomy's HAL/MAL (EFT), numerous device drivers, and other integration hardware and software.

It is worth noting that the platform is a very old airframe and, as such, "pushbutton design" was nigh impossible. Emulation of multiple layers of control logic, ranging from relay and switch logic of the 50s to intermediate levels of flight automation of the 90s, all had to be simulated to address several different airframe configurations. So test equipment characteristic 1, above, was impossible to achieve. Furthermore, the technologies associated with the Bloomy Simulation Reference System were still under development, and it took us several years to get to the point of being able to do a fairly close approximation of "pushbutton" (or scripted) design.

Because of the platform's long lifespan, many "unknowns" were discovered during the process of integrating the test systems with the units under test. Fortunately, the use of the largely-COTS Bloomy Simulation Reference System allowed for re-cabling to adjust the system to meet these previously-undiscovered requirements. Additional signal conditioning cards were added and cabled into the system, and though the system was the first of its kind ever built, the changes were made reasonably smoothly since they did not require digging through nests of wires. This "cable-level" technology was made possible because Bloomy is a test equipment company and invested millions of dollars and tens of thousands of man-hours into developing this and other test-centric products.

In rejecting the company's internal real-time test framework in favor of the COTS NI VeriStand and TestStand products, the company made two significant gains. First, no further development was required to the internal real-time test framework which has no capability to integrate with a requirements management system, much less the platform's digital thread. Given budgetary constraints usually associated with development of internal tools—namely they're usually developed with capital funds and improvements are difficult to justify since they rarely demonstrate a return on investment—and the decreasing number of personnel who are available to continue maintenance of the toolset, the minimal extra cost of the COTS software suite was well-justified.

Second, the company was able to take advantage of NI's Requirements Gateway product. The Requirements Gateway product maps requirements to test cases and coverage reports and enables test results to automatically inform the digital thread with the results of testing. Once the mapping of requirement to tests was created, test engineers were able to focus on creating test cases and making improvements to the product without worrying about requirements tracing. They were also able to input results of non-automated tests (those that required manual interaction with the system, for example) which were also mapped back to requirements. Finally, generating coverage reports was automated and prevented days of coverage analysis. The result was that test efficiency increased by *30%*.

CONCLUSION

Digital Transformation is still in its infancy, and the tools reviewed herein continue to evolve. Bloomy has witnessed a sudden uptick in the number of companies which are implementing some form of Digital Transformation. We are not, however, seeing very much consideration being given to how test systems are expected to fit into these initiatives. As a result, we expect that as more companies extend their Digital

Transformation initiatives, more emphasis will be placed on this integration. For new test systems to best be able to interact with the digital thread, digital twin, and models, COTS systems will offer the best way to achieve the test systems characteristics enumerated above, and that technologies like those offered by Bloomy will be an efficient means of achieving these goals.

Author contact: Bill Eccles Bill.Eccles@Bloomy.com

REFERENCES

1. https://techcrunch.com/2021/05/28/once-a-buzzword-digital-transformation-is-reshaping-markets/
2. https://savi.avsi.aero/about-savi/savi-motivation/exponential-system-complexity/
3. Alexander, Christopher, *Notes on the Synthesis of Form*, Harvard University Press, Cambridge Massachusetts, 1964.
4. https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2020:20200126_incose_iw_mbse_transformation_v1.pdf
5. https://discover.3ds.com/sites/default/files/2020-05/simplifying-complexity-through-model-based-systems-engineering-no-magic-case-study.pdf
6. INCOSE.org
7. See INCOSE.org for more information about the other aspects of MBSE and other ways in which digital transformation is being addressed in systems engineering.
8. E.g.: https://www.ni.com/en-us/about-ni/newsroom/news-releases/ni-completes-acquisition-of-optimalplus.html
9. US Air Force, *Global Horizons Final Report: United States Air Force Global Science and Technology Vision – AF/ST TR 13-01*, United States Air Force, 2013. Excerpted by https://www.digitalengineering247.com/article/a-digital-stitch-in-time/, excerpted from
10. Ibid.
11. Singh, Victor and Willcox, Karen E., "Engineering Design with Digital Thread," *AIAA Journal* 2018 56:11, 4515-4528.
12. Ibid.
13. Gothing, Grant, *Hardware and Measurement Abstraction Layers*, National Instruments, 2016.
14. https://download.ni.com/evaluation/core-test/HAL_MAL_Web.pdf
15. Author's photo