STAT COE Report-08-2021

Model Selection and Use of Empiricism in Digital Engineering

Nick Jones, CTR
Wayne Adams, CTR
Sarah Burke, PhD

23 July 2021

The goal of the STAT COE is to assist in developing rigorous, defensible test strategies to more effectively quantify and characterize system performance and provide information that reduces risk. This and other COE products are available at [www.afit.edu/STAT.](http://www.afit.edu/STAT)

STAT Center of Excellence
2950 Hobson Way – Wright-Patterson AFB, OH 45433

Table of Contents

Executive Summary

As engineered defense systems become progressively more complex, the Department of Defense (DoD) is leveraging modeling and simulation (M&S) in the design, development, and engineering of new capabilities. As new systems are designed to serve broader Joint missions in challenging situations, often M&S is the safest, most economical, and in some cases, the only means available to test systems under development. When appropriate Digital Engineering (DE) models are not readily available, programs will need to modify existing models or create new ones to fill the gaps, either in the form of emulators based on existing models or entirely novel models based on new data. In these situations, empirical data may be required to support the creation or updating of a model of the desired system, whether that system is another model or some real-world object or phenomenon. The application of Scientific Test and Analysis Techniques (STAT) to the collection of empirical data helps ensure that the data is a meaningful and faithful reflection of the system to be modeled and supports the creation of useful empirical models. This best practice outlines the process by which a DE program should leverage the DE environment to efficiently model systems by first leveraging existing resources on the Digital Thread, and then by using first principles and empirical data to build new models where needed. The process also outlines how and when to leverage empirical techniques, via the STAT Process (Adams et al., 2020), to model new systems and phenomena, improve the readiness of existing models, and create model emulators when appropriate.

Introduction

As engineered defense systems become progressively more complex, the Department of Defense (DoD) is leveraging modeling and simulation (M&S) in the design, development, and engineering of new capabilities. As new systems are designed to serve broader Joint missions in challenging situations, often M&S is the safest, most economical, and in some cases, the only means available to test systems under development. Furthermore, the 2018 DoD Digital Engineering Strategy reiterated the importance of models in DoD engineering, and established a goal for Enterprise-level Digital Engineering (DE) across the DoD to accelerate the pace of the acquisition lifecycle and alleviate stove-piping in technology development (Deputy Assistant Secretary of Defense for Systems Engineering (DASD(SE)), 2018). DE is an integrated digital approach to systems engineering that uses authoritative sources of system data and models as a continuum across disciplines to support lifecycle activities from concept through disposal. With a growing reliance on M&S for test and evaluation, in some cases before physical articles exist to be tested, it is critical to decision makers and developers to have a metric to understand the readiness of models that represent systems under development.

DE will facilitate cross-collaboration and the sharing of information between systems engineers working in different areas of the Enterprise and at different times. Programs developing new systems will be able to reuse knowledge gained by past efforts by utilizing digital models of previously engineered systems and phenomena. Additionally, programs will be able to plan system modeling efforts with an eye towards collaborating with the future. The use of composable models and common simulation architectures will enable models of future systems to be built quickly from re-configurable components and updated as the systems and their operating environments become better understood (Defense Modeling and Simulation Reference Architecture (DMSRA), 2020). As time progresses and the DoD implementation of DE matures, reusable first-principles models will dominate the model ecosystem. But for the early generations of DE, and in the future as new technologies emerge, there will be instances where models are needed which do not exist, are not validated, or are validated for at least one use but do not have the appropriate Model Readiness Level (MRL). The MRL is defined by the model's agreement in fidelity with authoritative referents over its scope of applicability to support a desired new use (Ahner et al., 2021). MRL components and other relevant terms in this best practice are defined in more detail in the glossary section found at the end of this document. Successfully achieving this vision depends on having a reliable method to develop useful, reusable models.

This best practices explains the process of model selection in the objective DE environment, as well as some of the remedies available when models do not exist or where existing models lack the necessary MRL. When appropriate DE models are not readily available, programs will need to modify existing models or create new ones to fill the gaps, either in the form of emulators based on existing models or entirely novel models based on new data. In these situations, empirical data may be required to support the creation or updating of a model of the desired system, whether that system is another model or some real-world object or phenomenon. The application of Scientific Test and Analysis Techniques (STAT) to the collection of empirical data helps ensure that the data is a meaningful and faithful reflection of the system to be modeled and supports the creation of useful empirical models.

Models in Digital Engineering

DE leverages tools such as Model Based Systems Engineering (MBSE) to provide clarity and continuity throughout system development via a model-based approach to representing system requirements and interactions, but extends the use of models throughout the systems engineering lifecycle by introducing several new model and data constructs (Madni et al., 2019). The Digital Twin is defined as “an integrated multiphysics, multiscale, probabilistic simulation of an as-built system, enabled by [the] Digital Thread, that uses the best available models, sensor information, and input data to mirror and predict activities/performance over the life of its corresponding physical twin” (DAU Glossary, 2015). A Digital Twin “may [also] consider one or more important and interdependent […] systems,” including interacting systems and environmental effects that are not part of the system being developed (Glaessgen et al., 2012). The Digital Twin can enable programs to quickly evaluate design updates or changes via simulation; thereby greatly reducing the development timeline.

Once the designed capability goes into production, the Digital Twin can continue to be updated with real-world performance and maintenance data collected from the actual systems, here referred to as Digital Telemetry. The Digital Telemetry enables the Digital Twin concept to be extended to create an individual digital representation of each fielded unit, useful for making manufacturing improvements, performance life predictions, and logistics planning. In the DE paradigm, model verification and validation are critical to individual engineering lines of effort and across the enterprise. Model verification and validation ensure that Digital Twins are realistic representations of engineered systems and that Digital Telemetry is used appropriately.

As mentioned in the definition of Digital Twin, the critical technologies used in DE must be enabled by a Digital Thread: an overarching computer architecture which acts as a repository for, and provides access to, models, data, metadata, and services. The Digital Thread will serve as a source of authoritative truth for the program, and in the long term, for the DE enterprise (DAU Glossary, 2015, Zimmerman, 2019, Zweber, 2017). The Digital Thread is the central architecture at the core of the Enterprise DE concept, and will provide access to a library of architectures, models, and data for Enterprise-level users and programs. It will draw on standards and policy for model composability, interfaces, metadata, simulation architectures, verification, and validation to store, manage, and regulate appropriate reuse of models and data across multiple efforts.

Future programs who use the Digital Thread will be able to see the pedigree, fidelity, and scope of applicability of previously developed models to determine if they are appropriate to use in new applications. The Digital Thread will provide a standardized, searchable repository for archiving models and data to allow future programs to easily find applicable models and prevent the need for them to duplicate the work of an earlier program. Stored models that follow the standards of the Digital Thread will not only be usable by a program, but may be used to validate new models based on either new data or first principles. The step change of keeping models on the Digital Thread potentially forever and providing enterprise users with structured access to them will eventually provide a library of referent models to validate against. The models in the library, if designed with model composability in mind, may even be combined to produce a more precise model with capability over a wider application range.

All program-validated models along with their associated test data and metadata should move to the Digital Thread, where they can be stored, managed, further validated, and combined. Any model on the Digital Thread, be it based on first principles, lookup from raw data, or empirical analysis, must be continuously evaluated and revalidated for each new use. This principle, called Continuous Validation, is described in Ahner et al. (2021). However, the Digital Thread will not come with a ready stock of models for every use case; the model library will have to be built up by the user community over time. Even once the Digital Thread is an established part of DE in the DoD, there will still be novel systems for which original models must be developed. In the absence of previous modeling and well understood first principles, empirical modeling will be the critical enabler for expanding the model library on the Digital Thread, and the DE capabilities of the DoD Enterprise.

Model Selection and Development in Digital Engineering
Models used in DE represent the physics, materials, objects, phenomena, and/or processes necessary to develop, test, maintain, and improve real-world systems and objects. Programs developing systems in the objective DE environment will use models in a digital test environment to evaluate whether or not the program's requirements can be met before expending the resources to physically manifest the system being developed. The flowchart in Figure 1 shows the process by which a DE program should leverage the DE environment to efficiently model systems by first leveraging existing resources on the Digital Thread, and then by using first principles and empirical data to build new models where needed. The process also outlines how and when to leverage empirical techniques, via the STAT Process (Adams et al., 2020), to model new systems and phenomena, improve the readiness of existing models, and create model emulators when appropriate. The scenarios a user may encounter when following Figure 1 are outlined in the following sections.

It is worth noting that most programs won't be developing completely new systems, and likely won't need to develop all new models to represent them. The most likely scenario for programs developing systems by the use of DE will be a hybrid approach that combines several of the scenarios below: 1) obtain as many of the models as possible from the Digital Thread, 2) reduce new model development by updating models from the Digital Thread, when feasible, 3) develop as many first principles as needed for program use, and 4) fill the remaining gaps with empirical models derived from testing.

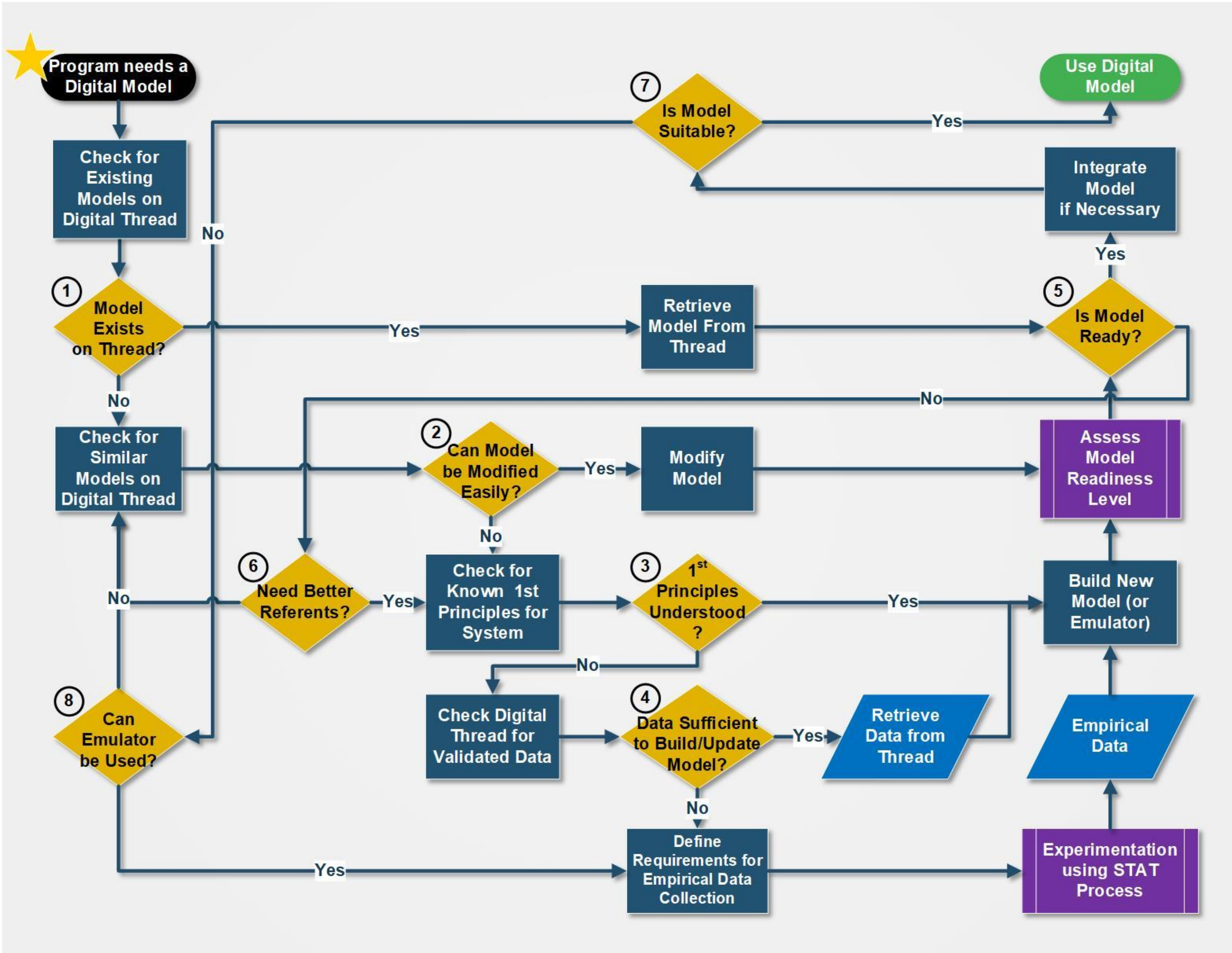

Figure 1: Model Selection and Empirical Testing in Digital Engineering

Scenario 1: Using a Model from the Digital Thread
In DE, programs seeking a model should focus their initial efforts on finding models on the Digital Thread that represent their systems and sub-systems. In some cases, every model a program needs may already exist on the Digital Thread. For example, a program's goal may be to improve an existing system so that it can operate in an expanded set of conditions by building it with different, but well-understood materials. The existing system will have an operational model, the subsystems may have interaction models, the different materials will have properties models, and there will be additional models to generate relevant environmental stimuli. In this first scenario, the program will acquire all the models it needs from the Digital Thread and, provided their MRLs are sufficient, use them directly. The program will then integrate the models to build a digital test environment to get data for analysis. This scenario is denoted in Figure 1 by the "Yes" path from the decision point labeled "1."

Scenario 2: Modifying Existing Models from the Digital Thread
Programs may not always be able to find the exact models they need on the Digital Thread. In this scenario, if a similar model is available, the program may be able to modify it to suit their needs, as shown by decision "2" in Figure 1. The model will be ready for use once the program has modified the model and has determined its MRL to be sufficiently high. Programs looking for models to modify should pay particular attention to the MRL, composability of the models on the Digital Thread, and any information documented in the model metadata. Not all models may be as easily modified as others. If a model requires extensive modifications to be useful (greater than 20% of the code needing to be modified) programs will likely find that it is a better use of their resources to build a new model.

Scenario 3: Building New Models
In the third scenario, the program may not find any pre-existing models on the Digital Thread that cover the needed system with the needed scope and fidelity, nor any models on the Digital Thread that are similar or easily-altered enough to be modified to represent the new system. Programs developing novel systems will likely encounter this situation for at least part of the overall model representation they need, even if models for some common system subcomponents are available on the Digital Thread. In this scenario, the program will have to develop new models using one or more of the model data sources outlined below.

New Models from First Principles and Existing Data. New models will ideally be based on a first principles understanding of the subject. First principles are built on a logical and/or physical understanding of the system, and therefore have some inherent trustworthiness. But first-principles models take time to develop, and showing model readiness often requires a real-world referent. As systems become more complex or the technologies and applications become more novel, the practitioners may not have the level of first principles understanding of the new system to create a useful digital model. In some cases, there may be known first principles that can support modeling part of a system, but that may not cover the entire system. If the known first principles don't support building a new model, then the program should check the Digital Thread for previously collected data, either from testing or operational use of sufficiently similar systems, as shown in the "No" path from decision "3" in Figure 1. Data retrieved from the Digital Thread will have a known pedigree (established as a prerequisite for the data to be on the Digital Thread), and thus may be trusted as the basis for a model. If such data is available, as in the "Yes" path from decision "4," then it can be retrieved from the Digital

Thread and directly used to support modeling the system.

New Models based on Empiricism. If there are no sufficient first principles or validated data from similar systems available on the Digital Thread, then the program must collect its own data to support the creation of an empirical model, as demonstrated by following the "No" path from decision "4." The first step in the creation of a new model is to define the desired model's requirements. The model's requirements should be well understood by this point in the process in Figure 1, since the requirements will be critical in determining if any of the existing models, data, or first principles are ready to support the program's needs. However, the program should re-examine the requirements for the model to ensure they meet the criteria of being Specific, Unbiased, Measurable, and "Of practical consequence," or "SUMO," as outlined by Coleman and Montgomery (1993). Once the requirements have been defined, the program can conduct the needed testing according to the STAT Process, as outlined in the next section. For example, it may be nearly impossible to build a model from first principles of the physical effects of an exploding fuel tank after the impact of a munition using finite element analysis (FEA) and computational fluid dynamics (CFD), due to the complex and dynamic nature of the system. In such cases, a series of empirical studies can reveal cause and effect relationships in systems more quickly than FEA and CFD models. Empirical studies using the STAT Process can simultaneously examine a broad range of factors, such as projectile velocity, direction, whether or not it is incendiary, fuel tank fill-level, and the interactions of those factors, to determine their effects on the system. Analysis of the data gathered can provide an empirical model that can be used to search for combinations that are most likely to cause a spreading fire or explosion.

Scenario 4: Model Readiness and Empirical Referents

Whether a program uses a model directly from the Digital Thread, modifies an existing model, or builds a new model as in the scenarios described above, the program should always assess the model's MRL to determine if it is ready for use. In all of these scenarios, if the MRL shows that the model is ready, then the model can be used. However, if the MRL is insufficient, then the "No" path from decision "5" feeds the program back into the process, but with different goals than previously discussed.

If the model is not ready, the program will have to determine why. One reason may be that the referents are insufficient in one or more aspects of the MRL: fidelity, scope, or referent authority. If the referents available are insufficient, a program can employ the same methods used for developing a new model in Scenario 3 to find referents that will improve the MRL, as in the "Yes" path from decision "6." In addition to acting as referents to show the readiness of newly created models, data from empirical studies may also serve as supplementary referents to increase the MRLs of models that are already on the Digital Thread. For example, one program may develop an original model (empirical or otherwise) for its own use case and then transfer it to the Digital Thread once it has a sufficient MRL. New programs wishing to reuse the model, as in Scenario 1, but needing to use the model in a different scope or at a different level of fidelity than the program that designed it, may need new referents to determine the model's MRL in the scope and fidelity of interest to them. Data from an empirical source can serve as a referent for model applicability in a new scope, improved understanding of fidelity in the existing scope, or provide a more authoritative picture of the system the model represents. Application of first principles, legacy data from the Digital Thread, and/or new empirical evidence can all serve as referents for a model previously developed or taken from the Digital Thread. However, in Figure 1 at decision "6," a program may also find that they have all the right authoritative referents to cover the needed scope and fidelity, but the model doesn't fit the referents well. In this case, the program must conclude that the model is a poor representation of the system, and return via the "No" path from decision "6" to the Digital Thread to revisit the process of identifying a modeling approach.

Scenario 5: Model Emulators

Once the program determines the model to be ready, the final decision before use is the determination of model suitability, decision "7" in Figure 1. While the MRL gives a metric of how well the model can be trusted based on its similarity to some authoritative referent(s) in scope and fidelity, model suitability deals with the usability of the model: how well the model can be implemented and executed in a digital environment. If the model has a high MRL (i.e., matches the referents well), but is unsuitable for technical or programmatic reasons (e.g., schedule, run time, or computational constraints), then the model cannot be used in its current form. In this scenario, the program may wish to build an emulator of that model to use in its place, as in the "Yes" path from decision "8." For example, the first principles CFD model discussed in Scenario 3 might take a day or more to run. If the run time is too long for the model to be useful, the program should follow the "Yes" path and return directly to the application of empirical methods via the STAT Process, as shown at the bottom of Figure 1. However, in this case the system to be empirically tested (and modeled) will be the existing CFD model. The program can build an emulator of this model, which will likely execute faster and with fewer computational resources than the original model of the same phenomenon, and will be more suitable for use. If the emulator produced has a sufficiently high MRL, it can then be used by the program in place of the original model. However, if the program determines that some property of the original model cannot be abstracted into an emulated representation, then the program must either find the resources to use the model as-is, or, as was the case with a poor-fitting model in Scenario 4, return to the process via the "No" path from decision "8" to identify a suitable modeling approach.

Note on Model Integration

The final step before model suitability assessment and use in Figure 1 is for the program to integrate the model with other models, if needed. In the DE of complex systems, system-level Digital Twins will likely be composed of multiple sub-models and integration will be a key step in their assembly and use. Model integration is a non-trivial activity which is beyond the scope of this best practice, but practitioners should understand that integration may require the skill and expertise of software developers, software integrators, subject matter experts for the systems represented by the integrated models (both separately and as an integrated system), and that the ease or difficulty of model integration can be greatly impacted by model complexity and composable design. More information on tools and methods for model integration can be found in the standards referenced in the DoD Modeling and Simulation Related Standards and Best Practices Guide (2010).

Building Empirical Models with the STAT Process

Building empirical models often requires testing to collect data. In Figure 1, this test phase is accomplished via the STAT Process, represented by a block located in the bottom right corner and entitled "Experimentation using STAT Process." The STAT Process (Adams et al., 2020), shown in detail in Figure 2, is designed to promote test rigor so that the correct data is gathered, the proper analysis is performed, and the risks associated with using the model produced are quantified and understood.

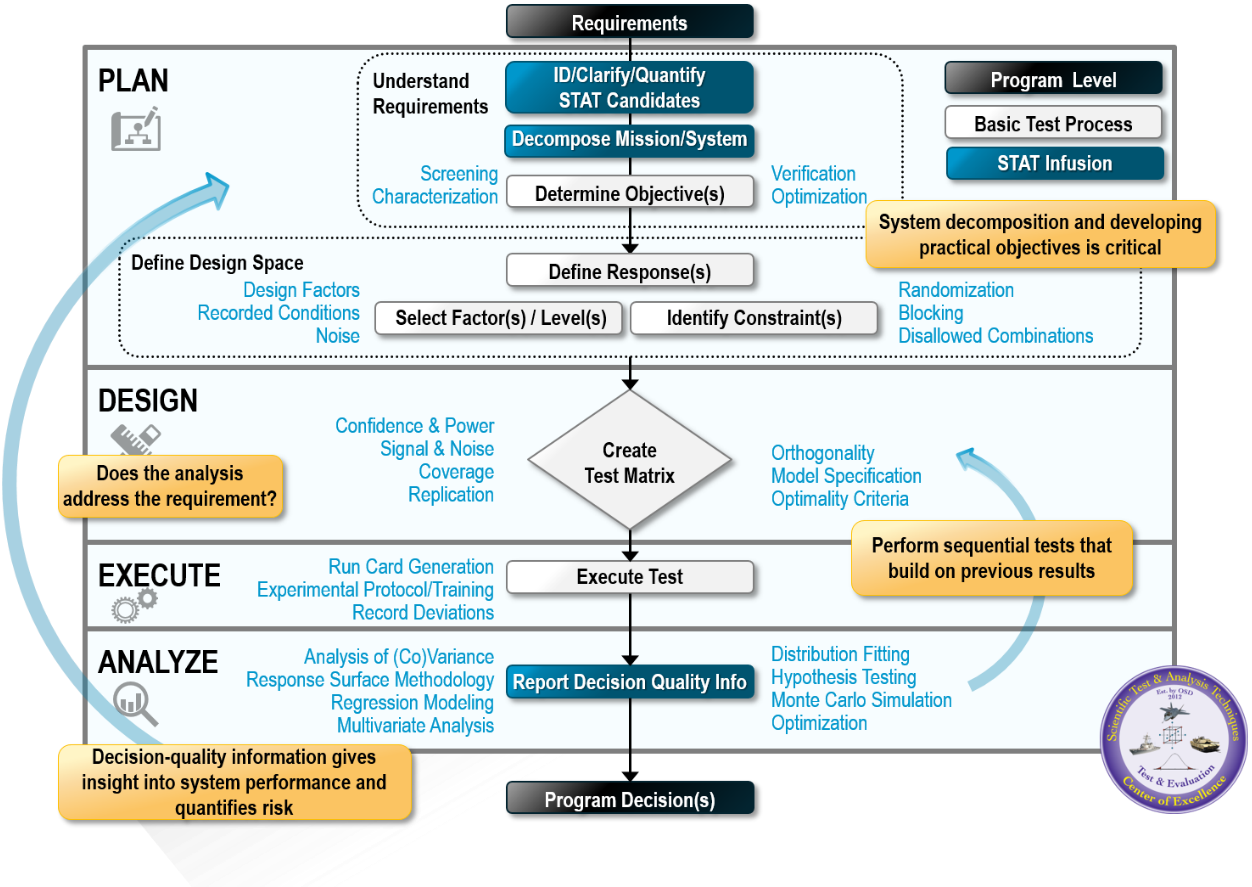

Figure 2: The STAT Process

The STAT Process is a framework for test and evaluation that leverages a methodical approach with rigorous test and analysis methods founded in statistics to efficiently provide the information needed to understand and model a system or phenomenon. Once an empirical model has been created, it is critical to employ proper validation methods such as those covered by Witten et al. (2016) to ensure the validity of the empirical model produced. When producing empirical models to fill gaps in a DE environment, the application of the STAT Process and the STAT tools and methods used will be the same as they would in any traditional test scenario. Whether the test is conducted on a physical article or an existing model to create a new model, a new referent, or an emulator, the application of the STAT Process is the same. The greatest distinction for empirical modeling in DE is in the definition of the requirements. STAT requirements for empirical modeling in DE should be determined from the characteristics needed to enable model readiness, usability, and composability.

Designing Empirical Models for the Digital Engineering Ecosystem

When initially developed for a single program, a model only needs appropriate fidelity and scope to address the program requirements. In order for one program's models to be reusable in support of other efforts in the DE environment, consideration will need to be given towards expanding the applicability of the model to new scopes and uses. If programs design for composability and model updating during the initial model creation, empirical data may be used to update the model or reassess its readiness later as new applications arise. Optimal algorithms for augmenting existing data sets are especially well suited to improving existing models (Goos, 2011). Scope improvements might include validating the model over previously unvalidated factor and response ranges, updating the model to cover wider factor and response ranges, or adding new factors or responses of interest to the model. New uses will include circumstances where a model originally designed for one program is applied across multiple programs. For example, a program might want to reapply a mathematical representation of water designed for testing naval vessels to modeling hydroplane resistance of tread patterns for wheeled vehicles. Instead of the water having ocean depth, as when working with ships, it can only be a few inches deep for testing of a wheeled land vehicle. The Navy may not have created the model with small depths in mind, but if modeled appropriately the surface-to-water interactions should remain sound. A thorough check as part of Continuous Validation will help to inform the tread-design team whether or not the water model can be used.

Programs building empirical models in DE should document the critical model properties in the model metadata. That metadata should then be carried forward to any system-level models that the model is integrated into. The metadata should include the nature and quality of information the model should accept as input and/or produce as output, any scope constraints that might limit the MRL in new applications, and any critical concerns needed to ensure compatibility in a composable model ecosystem. The Modeling and Simulation (M&S) Community of Interest (COI) Discovery Metadata Specification (MSC-DMS) covers a broad selection of the types of information needed to make a model discoverable in a DE environment and to support its appropriate reuse, with syntax examples (2012).

Conclusion

Empirical data and the models derived from them will continue to be an important part of how programs make decisions even after DE and the Digital Thread become the norm. Modeling the systems needed (physics, chemistry, economics, politics, etc.) to successfully meet program goals does not require a full first principles understanding of the world. Empirical models, based on well-reasoned selection of required outputs and a related set of inputs, provide the data necessary to make defensible decisions. DE will serve to extend the useful life of empirical models. In the future, as computational power increases, existing empirical models are likely to be replaced by more complex first-principles models. The old empirical model's role will change to becoming a referent used to validate the new generation of models. Empirical models can be designed to carry with them an understanding of how the modeled system/process works. This makes them a perfect tool for being reused and adopted by future programs.

Glossary

Accuracy - the degree to which a parameter or variable, or a set of parameters or variables, within a model or simulation conforms exactly to reality or to some chosen standard or referent (DoD M&S Glossary).

Emulator - a device, computer program, or system that accepts the same inputs and produces the same outputs as a given system (SISO-REF-002-1999).

Fidelity - a model's level of consistency with reality, defined in the four dimensions of accuracy, resolution, and repeatability.

Model - a physical, mathematical, or otherwise logical representation of a system, entity, phenomenon, or process (DoDI 5000.61, DoDI 5000.70).

Modeling and Simulation (M&S) - the use of models, including emulators, prototypes, simulators, and stimulators, either statically or over time, to develop data as a basis for making managerial or technical decisions (MSE M&S Glossary).

Referent - a codified body of knowledge about a thing being simulated (SISO-REF-002-1999).

Referent Authority - the strength of credibility of a referent's claim to be a high-fidelity representation of reality.

Repeatability - the similarity of the results obtained from the same model (or referent) over multiple observations under the same input conditions.

Resolution - the degree of granularity with which a parameter or variable can be determined (Pace, 2015).

Scope - the capabilities, limitations, and assumptions with respect to the inputs and outputs representing the relevant mission parameters, environmental conditions, constraints, requirements, and their mode of representation.

Simulation - a method for implementing a model over time (DoDD 5000.59, DoDI 5000.61, DoDI 5000.70).

Specific Intended Use - the set of dimensions, ranges, and assumptions of the model inputs and outputs needed to represent the modeled system's relevant mission parameters, environmental conditions, constraints, and requirements, combined with the additional constraints imposed by the target modeling environment and the required level of fidelity for the specific stage of program development.

Validation - the process that determines whether a model has sufficient fidelity relative to an appropriate referent(s) for a specific intended use.

Validity - a model's level of detail over a pre-specified scope relative to an appropriate referent(s).

References

Adams, W., Burns, D., Burke, S., Divis, E., Harmon, M., Jones, K., . . . Truett, L. (2020). Guide to Developing an Effective STAT Test Strategy V 8.0. STAT COE, Department of Defense. Air Force Institute of Technology.

Coleman, David E. and Montgomery, Douglas C. (1993). A Systematic Approach to Planning for a Designed Industrial Experiment. Technometrics 35: 1, pp. 1-12. doi:10.1080/00401706.1993.10484984.

Defense Acquisition University, (2015) Digital Twin. Glossary of Defense Acquisition Acronyms and Terms. https://www.dau.edu/glossary/Pages/Glossary.aspx#!both|D|27349.

Department of Defense. (2010). Modeling and Simulation Related Standards and Best Practices Guide.

Deputy Assistant Secretary of Defense for Systems Engineering (DASD(SE)). (2018). DoD Digital Engineering Strategy. United States of America, Department of Defense, Office of the Secretary of Defense.

Glaessgen, E. H., Stargel, D. S. (2012) The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, pp. 1–14, doi:10.2514/6.2012-1818.

Goos, P. & Jones, B. (2011). Optimal Design of Experiments: A Case Study Approach (1st ed.). Wiley.

Madni, A., Madni, C., Lucero, S. (2019). Leveraging Digital Twin Technology in Model-Based Systems Engineering. *Systems*, vol. 7, no. 7, 2019.

Modeling and Simulation Coordination Office (MSCO). (2020). Defense Modeling and Simulation Reference Architecture (DMSRA).

Modeling and Simulation Coordination Office (MSCO). (2012). Modeling and Simulation (M&S) Community of Interest (COI) Discovery Metadata Specification (MSC-DMS).

Simpson, J. (2014). Testing via Sequential Experiments. August. https://www.afit.edu/stat/statcoe_files/Testing via Sequential Experiments.pdf

Witten, I. H., Frank, E., Hall, M. A., Pal, C. J. (2016). Data Mining: Practical Machine Learning Tools and Techniques. Netherlands: Elsevier Science.

Zimmerman, P., Gilbert, T., Salvatore, F., (2019). Digital Engineering Transformation across the Department of Defense. *Journal of Defense Modeling and Simulation*, vol. 16, no. 4, pp. 325–38, doi: 10.1177/1548512917747050.

Zweber, J., Kolonay, R., Kobryn, P., Tuegel, E. (2017). Digital Thread and Twin for Systems Engineering: Pre-MDD through TMRR. *AIAA SciTech Forum - 55th AIAA Aerospace Sciences Meeting*, pp. 1–26, doi:10.2514/6.2017-0875.