CFD Validation:
Illustrations of Mutual Accountability and Validation Dialog throughout the Engineering Lifecycle

Nigel Taylor[*]

MBDA UK Ltd., Filton, Bristol, BS34 7QW, UK

William Schindel[†]

ICTT System Sciences, Terre Haute, IN 47803, USA

This paper is one of three being presented at this conference (AIAA SciTech 2024) considering how the principles of mutual accountability and validation dialog may be applied to address shortfalls in the current status of CFD validation when viewed in the context of the NASA CFD2030 Vision. By appealing to principles that have been refined and proven over many decades throughout the engineering and scientific communities, it is shown how the practices that have been developed specifically to support computational model validation may be viewed consistently with other means of accomplishing validation. In addition, the concept of a model validation landscape is introduced. This allows the various approaches that may be taken toward computational model validation – from the heuristic to the rigorously quantified – to be viewed from a consistent perspective. Specific opportunities that may arise as a result of the improved communication (validation dialog) afforded between stakeholders are also identified. A simplified analysis is presented to illustrate how fragmentation in the overarching model development lifecycle has contributed to the current shortfall in validated CFD capability. It is recommended that the principles and analysis described herein are refined and extended to apply to a much broader set of computational modelling scenarios. The scope of this endeavour should extend beyond CFD and beyond the aerospace sector. One of its more immediate goals should be the publication of a broader series of guidelines on computational model validation that serve the stakeholder community as a whole.

I. Introduction

From an engineering perspective, it seems reasonable to assert[1] that users of a computational model would like to know how accurate the outputs produced by the model are considered to be (and for this information to be evidence-based), not how accurate they ought to be (requirement-based). Moreover, especially for well-studied regimes, users would like to know this information for a suitably wide range of model inputs *before* they use the model so that the expected accuracy of the model may be compared with the accuracy requirements for the task at hand. Indeed, the above assertions form central pillars of the NASA CFD 2030 Vision [2], e.g., *“A single engineer/scientist must be able to conceive, create, analyze, and interpret a large ensemble of related simulations in a time-critical period (e.g., 24 hours), without individually managing each simulation, to a pre-specified level of accuracy.”*

---

[*] Capability Leader, Aerodynamic Tools and Methods, AIAA Associate Fellow.

[†] President, INCOSE Fellow and INCOSE Patterns Working Group Chair.

Judging from the published literature on the subject, the current state of the art regarding CFD validation falls some way short of that required to realize the NASA CFD 2030 Vision. To quote from the most recently issued ASME Standard for verification and validation in CFD[3]:

> *"The objective of this Standard is the specification of a … validation approach that quantifies the degree of accuracy inferred from the comparison of [computed] solution and [measured physical] data for a specified variable at a specified validation point. … Consideration of the accuracy of simulation results at points within a domain other than the validation points (e.g., interpolation/extrapolation in a domain of validation) is a matter of engineering judgment specific to each family of problems and is beyond the scope of this Standard."*

Industrial concerns about focusing CFD validation approaches on single "validation points" (not to mention their scope being applied to single output variables) have been expressed for many years (see e.g. [4]). However, in contrast with other technological areas that have seen considerable improvements since the initial publication of the NASA CFD 2030 Vision, there has been no substantive change in the scope of the published guidance or standards on CFD validation over the last ten years.[5]

The AIAA Applied Aerodynamics and Ground Testing Technical Committees have recently formed a joint Focus Group to help facilitate more extensive interactions between the computational and physical testing communities. Named ICE - Integrated (Digital and Physical) Collaborative Experimentation - one of its first activities has been to address the situation outlined above with regard to CFD validation in the context of the NASA CFD 2030 Vision by appealing to the principles of mutual accountability and validation dialog.[6]

Following oral presentations delivered by the principal author to last year's AIAA SciTech and AVIATION Forums, this paper is one of three being presented at this conference on this subject area - see also [6,7]. Its focus is on establishing a framework by which we can better understand the role of validation in general – and CFD validation in particular – and the value of the interactions between the various actors involved. In Section II, the subject is introduced by outlining some of the fundamental concepts governing the practice of Engineering Discipline. Building on these, Section III contains a brief, simplified analysis of the situation outlined above regarding the status of CFD validation with respect to the NASA CFD 2030 Vision. Then, in Section IV, some of the benefits to be gained by fostering improved mutual accountability and validation dialog are identified. A new concept – that of a model validation landscape – is introduced to help facilitate this. Finally, a succinct set of closing remarks and recommendations is provided in Section V.

II. Fundamental Concepts and the Practice of Engineering Discipline

Problem abstraction and decomposition are vital aspects of engineering practice: the ways in which these subjects are approached will determine the processes used to engineer solutions to the task-at-hand, whatever that may be. In this Section we review some of the fundamental features of the frameworks that have been established – and proven – over many decades to provide guidance, facilitate managed delivery (to time cost and quality) and to ensure that the outcomes are endorsed by all stakeholders. We refer to the adoption of and adherence to these standards as the practice of Engineering Discipline.

"Discipline" might suggest a static landscape, but actual engineering practice is advancing continually, offering new candidates to prove themselves and find their way into standards and wider use. As befits this developing world, the international standards in which these frameworks are embedded are reviewed and updated on a regular basis. Therefore, we approach the subject from two perspectives. A succinct review of the salient features of current practice is provided in Section IIA (under the heading Engineering and the Engineered Product Lifecycle) and the principles underpinning one avenue of thought being pursued in the vein of continuous improvement are introduced in Section IIB (under the heading The Use of Shared Model-Based Patterns) The topics raised are extremely broad and have extensive supporting literature. Consequently, the material presented below is intended only as a succinct introduction to those features the authors considered most salient to CFD Validation in the context of the NASA CFD2030 Vision.

The concepts, standards, and references for these two sections draw primarily from descriptions of (1) engineering of systems of all types in general, for the most widely-applicable points; (2) engineering of software-based systems or subsystems, for that important but more specialized case; (3) engineering of computational models, still more specialized; and (4) engineering of the models used in CFD. Note especially that the underlying principles of Verification and, Validation, including the Quantification of Uncertainty, appear in all four cases, in spite of sometimes distracting differences in their uses of related terminology.

A. Engineering and the Engineered Product Lifecycle.

An explicitly declared underlying principle of the (engineering) standards framework is the requirement for application-specific interpretation, to address details that are specific to particular application scenarios. For CFD, this could include addressing the requirements associated with solving non-linear partial differential equations, or deriving suitable constitutive relations for the fluid being modelled, for example. Importantly, while local customisation is expected (required, even), this should not be accomplished at the expense of compliance with other aspects of the overarching framework: this would risk failure or problems being encountered elsewhere in the engineering lifecycle. Balancing such application-specific interpretations and local customisation with standard frameworks might at first sound potentially subjective or "mushy", but takes on greater formal meaning and ultimately a discipline-capable clarity and rigor in Section IIB.

For the current discussion, an "engineered product" is any entity whose definition, production, distribution, utilization, sustainment, or eventual retirement require "engineering processes" (identified below) that consume and produce information supporting each other as well as other "life cycle management processes" (also identified below).

Engineered products, which generally interact with other engineered products as well as natural systems, thus include aircraft, air bases, ground vehicles, communication systems, medical devices, consumer products, service-providing business systems and computational models, including those used in CFD.

Even though most engineered products are not alive in the biological sense, all of them have "life cycles", often years in length, over which the engineered products are defined, produced, distributed, used, maintained, improved or otherwise revised, and retired. Although engineering processes might be assumed to be associated primarily with the "definition" portion of those life cycles, it turns out that engineering processes interact with and are essential to the success of other, non-engineering processes. These include product production, distribution (including marketing, logistics, installation), utilization, sustainment, update, and retirement.

A typical representation of the range of life cycle management processes, described from an engineering point of view that includes non-engineering processes interacting with engineering processes, is the systems engineering "Vee Diagram" of Figure 1, and the related ISO15288 descriptive standard [8]. It summarizes both engineering and non-engineering processes across the life cycle, but especially illustrates the fact that many of them are organized around different levels of hierarchical decomposition of the engineered product into its inter-related parts or subsystems, each requiring attention of different instances of the management processes.

The Engineering Processes consume and produce *information*. Figure 1 illustrates the idea that the Engineering Processes consume information from (and supply information to) each other and non-engineering processes. That information itself is not emphasized in this diagram, which is more concerned with the implied inter-dependencies of the processes. By contrast, Figure 2 introduces a more balanced view that includes the information consumed and produced, and begins to suggest the complexity of what is occurring. Of particular importance is the idea that the different "information segments" shown in Figure 2, produced by separate processes, may turn out to be inconsistent with each other. For example, the design of a system, subsystem, or component may violate some requirement that is allocated to it; a formalized requirement may be inconsistent with a stakeholder need; a constructed or purchased component may be inconsistent with its specified design, the use or maintenance of a delivered product may be inconsistent with its specified requirements, etc.

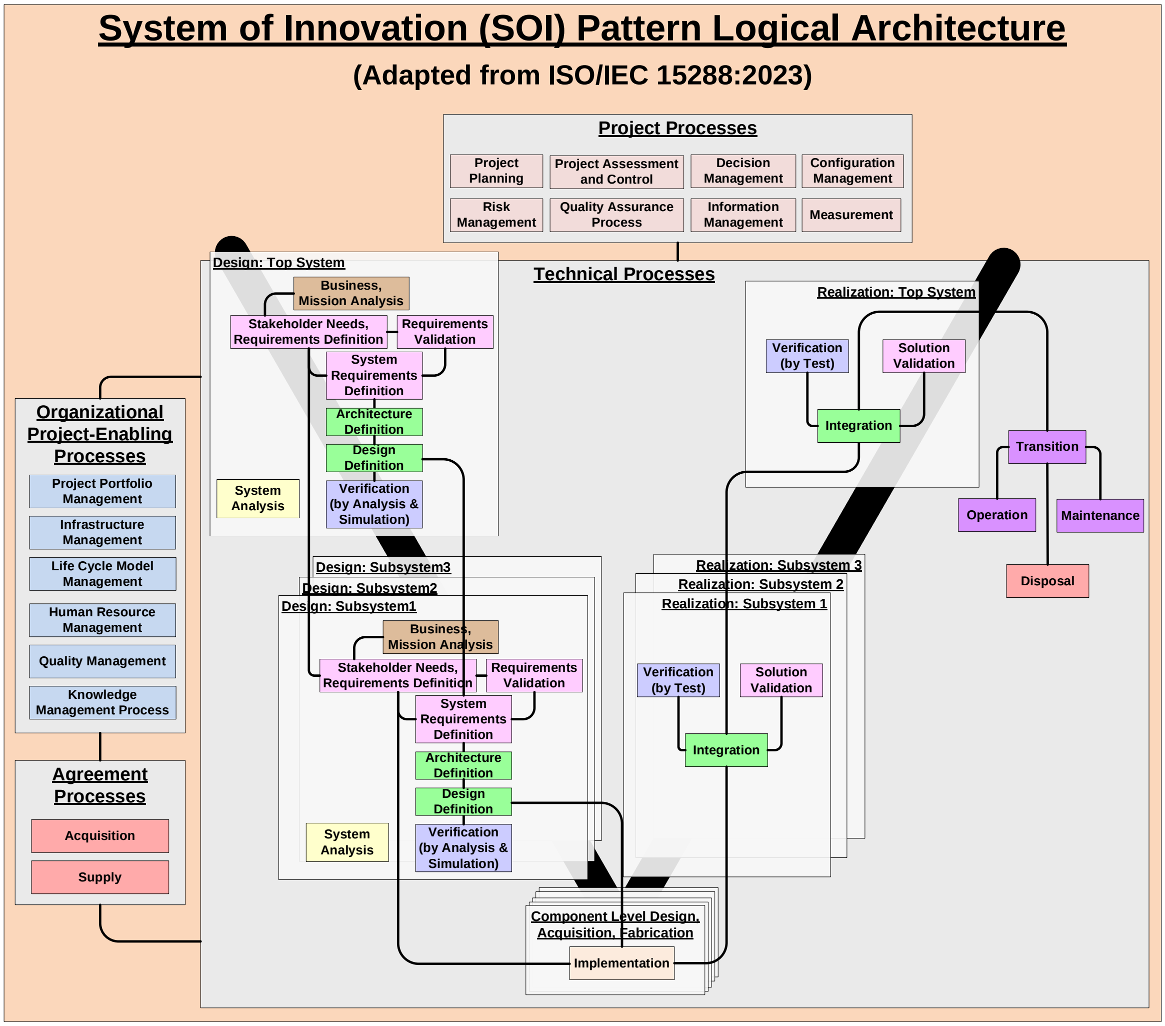


Figure 1: The Systems Engineering "Vee" Diagram, Showing ISO15288[8] Life Cycle Management Processes

This introduces the idea that the current *consistency state* of the information segments at the bottom of Figure 2 is critically important, and can vary across the life cycle of the engineered product. The most important "state" of the overall life cycle management processes is the state of the information at the bottom of Figure 2, in configuration space—and especially the state of the inter-segment consistencies. For this paper, the validation state of a CFD model is a prominent example of such "managed consistencies".

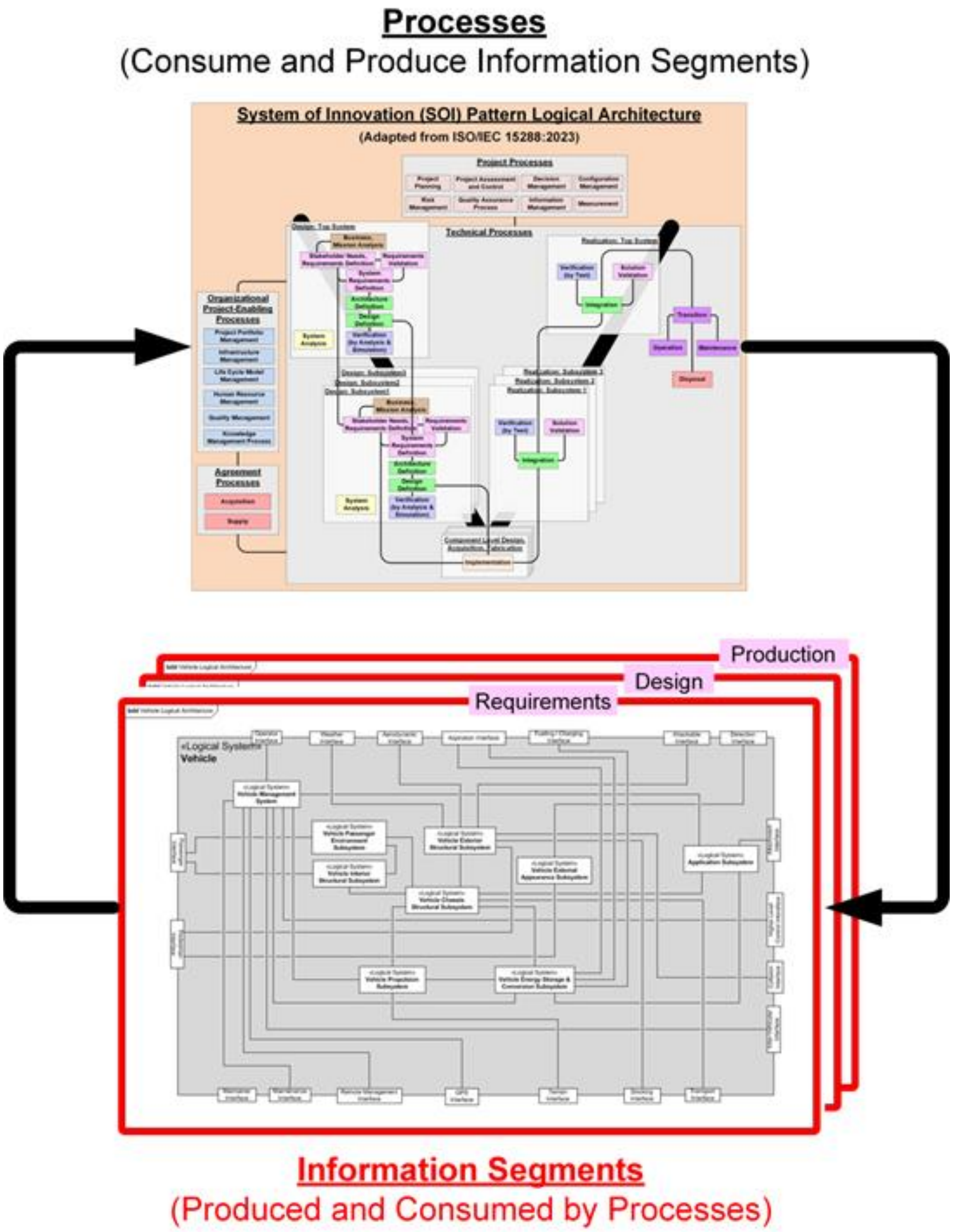


Figure 2: Visualizing the State of Information and its Consistency

Developed over many years and widespread efforts, individual standards, conventions, and recommendations for certain of the related processes include those for systems in general [8,9], software in particular [10], and even more specifically for computational models [11] and CFD models [12]. An essential simplifying overview relevant for this paper is:

1. **Consistency Paradigm**: Virtually all of the efforts of the "standard" life cycle management processes summarized above are about creating a "consistency thread" of information segments spanning the various traditional and well-identified information segments, for which the degree of consistency and degree of confidence in that consistency are both known through evidence. This is the "consistency management" perspective further described in the recent AIAA reference models for Digital Threads [13] and Digital Twins [14].

2. **Criticality of Rigor**: When we refer to “rigor” of the related engineering disciplines, we are not just referring to technical details of individual disciplines, but to the care with which the above cross-discipline web of evidence-based consistency is known and managed. Mutual accountability requires rigorous accounting first.

3. **Impact of Spread of Model-Based Representation**: For much of the history of engineering, only limited aspects of the information described at the bottom of Figure 2 have been subject to representation using the explicit models of STEM, such as those found in fluid science and CFD. Many other aspects of system requirements, stakeholder needs, production processes, testing, operation, and other aspects have historically been described by written prose for human interpretation. It has been commonplace for three human readers of a prose requirements document to have (at least) three different interpretations of it. However, if three electrical engineers had different interpretations of an electrical schematic, they would stop work and consider it remarkable. The recent spreading of more general model-based methods (e.g., Model-Based Systems Engineering, or MBSE) across this whole landscape means that CFD information will not just be a model-based “island”, but joined up with other model-based representations.[15] This has profound implications for the ability to detect and manage consistency threads, including the use of automated assistants, and is further examined in Section IIB.

4. **Learning, Uncertainty and Risk**: With the above strong emphasis on the acquisition of managed confidence information webs, it becomes clearer that what is being described is group learning, with the resulting web of information being what is learned by the enterprise, and the opportunity to manage that consistency using explicitly managed uncertainties, potential impacts, and estimations and management of resulting risks. This throws open the methods and literature of learning, including both human and more recently machine learning, risk management methods and representations, and Design of Experiments (DOE) for the larger domain of innovation systems as a whole [16,17].

5. **The Importance of Cross-Domain and Cross-Discipline Semantics**: Practitioners in different disciplines, including computational modelling, have terminology for their work and work products that is sometimes inconsistent with those of other disciplines (or even other practitioners in the same discipline), as observed in Chapter 2 of[11].[*] With the growing potential of model-based methods referenced above and model-based patterns discussed in Section IIB, spanning this language divide becomes more critical. Such an endeavour would be a natural role for AIAA and other technical societies such as INCOSE, ASME, IEEE, NAFEMS, and others.

6. **Mutual Accountability and Validation Dialog:** A special case of (5) is this paper’s concern—the recognition that mutual accountability and validation dialog between stakeholders are both crucial to progressing through the lifecycle. Key to team-building, the ability to communicate across disciplines is essential to both mutual accountability (for expressing the cross-discipline consistency of our respective information segments) and validation dialog (for the efficient reconciliation of detected inconsistencies which will always arise) In seeking to team-manage the consistencies, an important by-product of mutual accountability and validation dialog is that the combinatorial aspect of shared ideas stimulates new thinking and approaches to addressing challenges – and thereby reconciling inconsistencies at a higher level of performance than previously imagined. This is further discussed in Sections II.B, III, and IV.

7. **Difference Between Decomposition Hierarchy versus Order of Work**: Vee diagrams such as Figures 1 and 2 have sometimes been interpreted as implying order of work processes, such as proceeding from the left top to bottom (sometimes referred to as a “waterfall” interpretation), and then from the bottom to right top. However, this need not be the case. Time sequence is not the deepest significance of these diagrams (which are about the hierarchical interdependence of information segments they consume and produce). Experienced engineers quickly realize that work can and does flow in both directions. For example, inconsistencies between requirements and designs are not always resolved by changing designs—sometimes requirements are relaxed. Even if the work is pursued somewhat “top down” linearly, it is necessary to iteratively perform updates of the cycle until inconsistencies are resolved. A more general interpretation is that (depending on project methodologies and

---

[*] This situation can be compounded by disparities in how the terminology is adapted to reflect the various transitions that occur during the engineering lifecycle. For instance, validat<u>ion</u> is considered a process; once the process is completed, the item subjected to it is described as being validat<u>ed</u> (subject to the limitations and constraints of the validation activity – the significance of which may, in certain circumstances, change with time). Similar comments apply to other processes (e.g. verification, yielding the status verified) and sub-processes (e.g. as part of a validation activity, the process of uncertainty quantifi<u>cation</u> yields quantifi<u>ed</u> uncertainty).

controls) the processes could be executed in parallel or in various orders of incremental update. In general, the decomposition hierarchy of the diagram describes interdependence of information segments, not the order of their update.

8. **Life Cycles and Engineering of Models versus those of the Systems That They Model:** Consider Figure 2 for the example case of an engineered aircraft system. The left side of that diagram shows "Verification by Analysis and Simulation" at multiple levels of decomposition of the aircraft. These may include use of CFD or other computational models to support the development of related aircraft designs. However, what if one of those simulations is itself a commercial information technology product that is used by many companies and projects, and has its own engineering product life cycle? In those cases, the simulation product itself will have its own "Vee diagram", potentially including its own decomposition, verification, validation, etc. This is an area in which the engineering community and computational modelling community might be seen as colliding in their terminologies. However, it is also possible to formally coordinate these by recognizing that the underlying concepts of consistency management are about Nature and not human convention, allowing mapping of different human-preferred semantics onto that consistent scaffolding.

9. **Validation in the Sense of ISO15288:** The process of Validation appears in more than one place in the ISO15288 framework. In each case, its essential purpose is "*the confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled*".[8] In Figure 1, "Validation" appears on both the Vee left side (as in Requirements Validation) and on the Vee right side (as in Solution Validation). The spirit of both of these is that validation is confirmed by "reaching all the way back to" the interests of the system's stakeholders:

    a. **Requirements Validation:** Per ISO 15288 6.4.11.1, Note 2: "Validation is also applicable to the artefacts (e.g., requirements, …) produced in the definition and realization of the system." Per 6.4.3.3.C.3, Note 18, ". . . This is one application of the validation process applied for the specific requirements." This is asking, from the stakeholder point of view, whether we have the right system requirements to drive the realization of the system.

    b. **Solution Validation:** Per ISO 18288 6.4.11.1, the purpose of this validation is " . . . to provide objective evidence that the system, when in use, fulfils is business or mission objectives and stakeholder needs and requirements, achieving it intended use in its intended operational environment." This is asking, from the stakeholder point of view, whether we have successfully realized and delivered the system needed by the stakeholder.

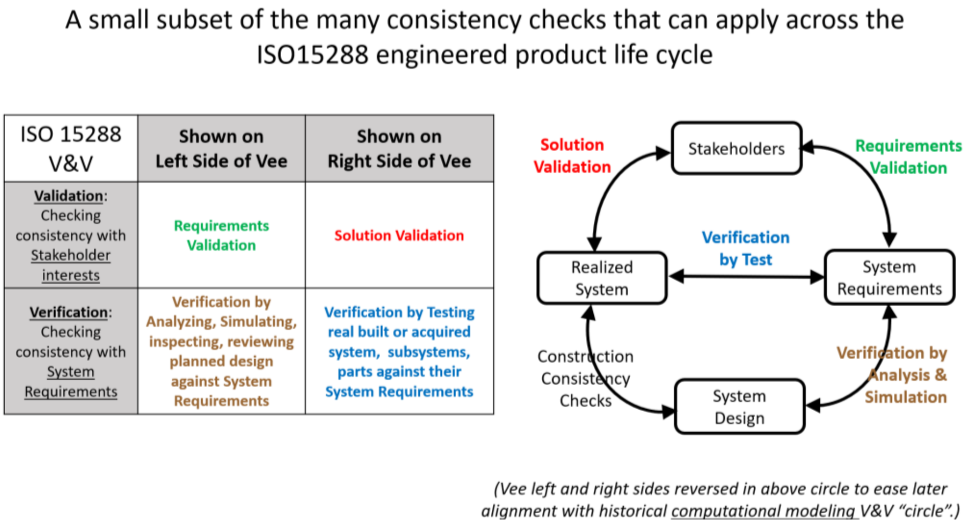
10. **Verification in the Senses of ISO15288:** Similarly, Verification appears in more than one place in the ISO15288 framework. In Figure A-1, "Verification" appears on both the Vee left side (as in Verification by Analysis and Simulation) and on the Vee right side (as in Verification by Test). The spirit of both of these is that verification is checking the consistency of the system's (planned or already implemented) design against the "only part way back but objective" system requirements. (Whereas validation, above, recognizes the primacy of stakeholder interests, it is also a fact that these are often expressed in less than objective or complete form, so System Requirements enter the picture as the consistency check anchor for Verification. There is thus a balancing act between stakeholder primacy and system requirements objectivity and completeness, and that is why "both Vees" are together so important.)

    a. **Verification by Analysis and Simulation:** Verifications on the left side of the Vee diagram often (not always) occur at times before the engineered system has been acquired or fabricated, thereby offering the promise of discovery of design problems (not meeting requirements, sub-optimality, alternatives and trade-offs, etc.) in advance of time and expense of building the real engineered system or its subsystems or parts. Accordingly, verification of this type relies of virtual simulation, "paper" analysis of planned design, inspection by peer to compare to past experience, analysis against known patterns of experience (see Section IIB), etc. This is an "early warning system" of the Vee.

    b. **Verification by Test:** Verifications on the right side of the Vee diagram occur after the engineered system (or subsystems, parts) have been acquired, fabricated, or integrated, providing the possibility of testing these real elements against system requirements allocated to them.

11. **A Simple Example Overview of ISO15288 V&V:** The above (items 9 and 10) discussion of ISO15288 V&V is summarized even further by Figure 3, whose right side introduces the "circle" representation in preparation for comparison to certain consistency management aspects of computational model community V&V. Notice that

this form of diagram conveys the idea of pair-wise "consistencies" comparisons. The comparisons shown here are not the only cases of V&V that are covered by ISO15288, but they provide initial examples for this discussion:



A small subset of the many consistency checks that can apply across the ISO15288 engineered product life cycle

ISO 15288 V&V	Shown on Left Side of Vee	Shown on Right Side of Vee
Validation: Checking consistency with Stakeholder interests	Requirements Validation	Solution Validation
Verification: Checking consistency with System Requirements	Verification by Analyzing, Simulating, inspecting, reviewing planned design against System Requirements	Verification by Testing real built or acquired system, subsystems, parts against their System Requirements

(Vee left and right sides reversed in above circle to ease later alignment with historical computational modeling V&V "circle".)

Figure 3: Some Key Elements of ISO15288 V&V, Derived from Items 9 and 10 Above

12. **Verification and Validation in the Senses of the Computational Modelling Community**: As discussed in [18] and [11] there are a number of historical interpretations of computational model verification and validation. Since these are not all consistent with each other, Figure 4 is offered as an early-published overview representation of computational verification and validation, from [19]. Over later years, more aspects of computational model V&V have been added, but this early representation has the virtue of simplicity for this discussion, as it summarizes some of the main and continuing ideas that span both scientific and engineering computational models. The parenthetical observations in b, c, and d below will become important when we view the computational model as an Engineered Product in its own right:

    a. **Conceptual Model versus Computational Model versus Real Modelled System:** In Figure 4, a computational model (the "Computerized Model") is the executable intended to provide an automated simulation of some aspects of a Real Modelled System. A finite element CFD model (including meshes, input output, and algorithms) of a real aerodynamic system (e.g., flow around an airfoil) is an example. The computational model is to be based on a Conceptual Model, such as a mathematical physics-based theoretical model based on air flow equations. Historically, these models were frequently assumed to be quantitative (numerical), although our engineering interests include additional forms of behavior, such as discrete or phase change behavior.

    b. **Model Verification:** In Figure 4, this refers to the consistency checks carried out to determine how well the Computational Model agrees with the Conceptual Model, for domain ranges of interest. Note especially that this does not tell us how well the Conceptual Model compares to the Real System, but provides a baseline supporting comparison. (Notice how that observation is similar in some ways to 10(b) above, in which checking consistency of implementation against requirements does not tell us anything about whether the requirements are valid.)

    c. **Model Qualification:** In Figure 4, this refers to the consistency checks carried out to determine how well the Conceptual Model agrees with the Real System, for domain ranges of interest. (This was the label applied in the 1979 reference for an assessment of conceptual model suitability—we are not concerned here with the specific name.) (Notice how this check on the conceptual model's validity is similar in some ways to 9(a) above, in which checking consistency of requirements with stakeholder interests helps affirm those requirements as an intermediate technical baseline for later downstream comparison checking.)

    d. **Model Validation:** In Figure 4, this refers to the consistency checks carried out to determine how well the Computerized Model agrees with the Real System it is expected to simulate, for domain ranges of interest. (Notice how this check on the implemented model's validity is similar in some ways to 9(b) above, checking the consistency of the delivered product against certain expectations.)

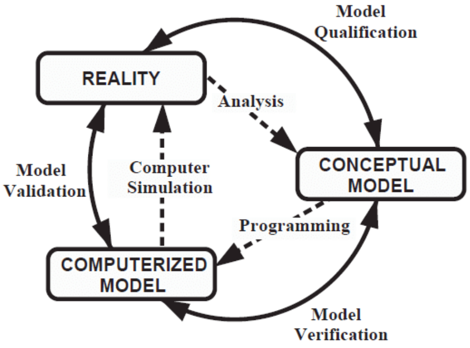


Figure 4: An Early View of Computational Modelling V&V Consistency Relationships (from [19])

13. **<u>Bridge from ISO15288 to Computational Model V&V</u>:** It is sometimes reported [11] that the engineering community's and the (scientific or other) computational modelling community's uses of the terms "verification" and "validation" are in conflict with each other. Nevertheless, item (1) above should help us see that some form of mapping or other reconciliation of these communities' descriptions is essential to a well risk-managed future in which project computational modelling information segments are readily checked for consistency (and reconciled where inconsistent) with larger information segments. That is the impetus for action, but the gap to be closed here may be smaller than perceived, for the following reasons. If there is disagreement at all, it is probably not that the various specific segment-to-segment consistencies are important candidates to monitor and manage for certain models and applications, but simply what names are used for each of these consistency checks. Using the perspective of Item 8 above, a computational model can be itself perceived as an engineered product subject to ISO15288. In that case, Figure 5 is a combined perspective. Note especially that this diagram is restricted to the earlier historical ontology of Figure 4; a recommended future mapping effort across the communities would include the enlarged ontology of current computational model life cycles.

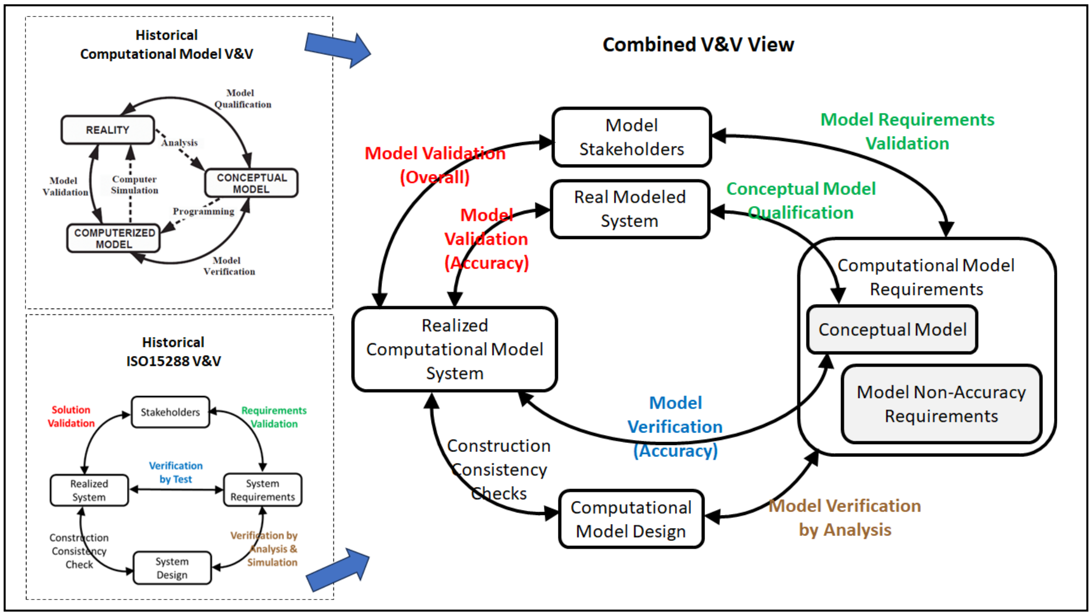


Figure 5: A Combined View Consistent with ISO 15288 [8] and Computational Modelling V&V

B. The Use of Shared Model-Based Patterns

Shared model-based patterns are at the heart of the impactful history of science and engineering, and particularly processes of group collaboration in discovery and representation of natural and human-engineered phenomena. Patterns in general are recurring regularities, observed and recognized (formally or informally) across time, space, or other indices. Patterns are expressed in different forms, but model-based patterns emerged in the last three centuries as the effective language of science and engineering [20,21]. Accelerated by the emergence of high speed computing machines, computational model simulations frequently represent quantitative aspects of that knowledge. The qualitative structure of that knowledge is likewise represented in the semantic (ontological) structure of system models (i.e., MBSE models). Model-based patterns have both fixed (recurring) aspects, which is how we recognize them, and variable (configurable, parameterized) aspects, which describe how different instances of the same pattern vary, and can account rigorously for allowed variabilities.

Shared model-based patterns are harnessed by current joint projects by members of AIAA, INCOSE, ASME, and NAFEMS in development of guidelines for planning aerospace digital threads and digital twins, along with managing the uncertainty of models in advanced manufacturing, using principles also applicable to other domains.[13,14], [22,23] Four kinds of configurable model-based patterns relevant to the use of computational and other virtual models are illustrated by Figure 6, numbered there and described as follows:

1. **S*Metamodel**: A generic STEM-based metamodel that defines a neutral, sharable semantic framework that can be mapped to numerous specific languages and tools (standards based, third party, etc.), while providing a common semantic bridge between them. The Systematica Metamodel (or S*Metamodel—see Figure 7), is used by the INCOSE MBSE Patterns Working Group to provide such a neutral foundation. It was developed and used over two decades in answer to the question "what is the smallest model content necessary to cover information required across the life cycle of systems, and to support model-based patterns?" Any model that satisfies (can be mapped to) the S*Metamodel is referred to as an S*Model, no matter what modeling language or tooling is involved. The need for such a shared framework was learned by observing that even models in standard languages were (a) often being generated missing basic essentials (too small) but also redundant and inconsistent with themselves (too big) at the same time; (b) incompatible with each other across projects, tools, teams, and supply chains; and (c) missing some of the basic lessons of three centuries of STEM. [24,25] The S*Metamodel also provides the foundation for configurable model-based patterns, which are S*Models of families of systems, phenomena, products, or processes, and are called S*Patterns. Refer to Figure 6. In the case of computational models, the Conceptual Model could be an S*Model (see Figures 4, 5), and if we want to cover a broad range of computational model configurations and their validations, the formal configurability of an S*Model can provide a supporting configurability discipline. The formal Interaction portion of the S*Metamodel provides explicit model-based context for the Phenomena identification of computational model PIRTs – see Figure 8.

2. **Model Characterization Pattern (MCP):** Not all models can be expected to be S*Models or otherwise uniform. However, under the already existing methodologies of scientific and engineering modeling, all models do require a degree of descriptive documentation. For computational models, examples of this are the model's intended context of use and the Phenomena Identification and Ranking Table (PIRT). For models in general, the style and content of model documentation was observed to vary greatly across different types and instances of models, model domains, model authors, and enterprises. While expecting the models themselves to all be consistent may be too much, it is plausible (and valuable) that their descriptions might be generated as configurations of a generalized metadata pattern, and that is what the MCP is. There are also other emergent examples of metadata for models [26,27]. MCP is a general model metadata S*Pattern that is configurable to describe any model—including but not limited to computational models. This provides information about the computational model, as a kind of "model wrapper" of key information supporting ability to share the computational model on an informed basis. Among other things, this metadata describes quantified uncertainty about the computational model, and its validation provenance.

3. **Domain-Specific Patterns for a Family of System, Products, Processes, or Phenomena**: It is in this space that tens of thousands of computational, descriptive MBSE, and other models have been and continue to be generated—but model-based patterns are interposed here to gain leverage and discipline. As shown in Figure 6, configured S*Models can be semi-automatically generated from the S*Patterns created to support them.

The variability of these patterns is formally encoded in the pattern's configuration rules driven by Stakeholder Features of each S*Pattern (see Figure 7) [28]. These models may represent systems and subsystems at any level of the Vee diagram of Figure 1 and, for validation purposes, the Model-Validation Hierarchy (MVH) [11]. Model-Based S*Patterns have been used to describe aerospace, ground vehicle, medical device, consumer product, advanced manufacturing, and other systems of many scales. [22]

4. **The Agile Systems Engineering Life Cycle Management (ASELCM) Ecosystem Pattern:** The models generated from patterns of type (3) above might be mostly models of manufactured products. In contrast, the ASELCM Pattern is used to generate models of the Innovation Enterprise itself—its engineering, production, distribution, support, or other processes. Of particular interest to the INCOSE Patterns Working Group is the use of this pattern as a neutral (descriptive, not prescriptive) reference for the description of various engineering and other business processes that themselves make use of models and patterns. AIAA, INCOSE, and other technical societies are actively engaged in advancing the state of engineering practice, with computational models and model-based patterns only part of a wide range of innovations to the engineering process itself. As these advancements are pursued by enterprises, it is generally not enough to just take a few educational classes if the enterprise is pursuing multiple time-urgent efforts. Historically, most descriptions of these enterprise processes have been in the form of prose-based handbooks, standards, and company policies, procedures, and work instructions. As the process becomes more complex or change-intensive, systems engineers realize that representing the engineering process (and other connected life cycle processes) as a system in its own right becomes more important. As shown in Figure 9, the ASELCM Pattern (itself a formal SysML MBSE Pattern) does so by representing the engineered products as System 1 and the system of engineering and life cycle management as System 2. This is the basis of the recently published AIAA Reference Models on Digital Threads and Digital Twins, which can be consulted for details [13,14]. The Consistency Management paradigm it describes is central to not only model validation but also the integrity of the entire life cycle under virtually any engineering methodology.[29] A few examples of such consistencies were illustrated in Figures 4 and 5.

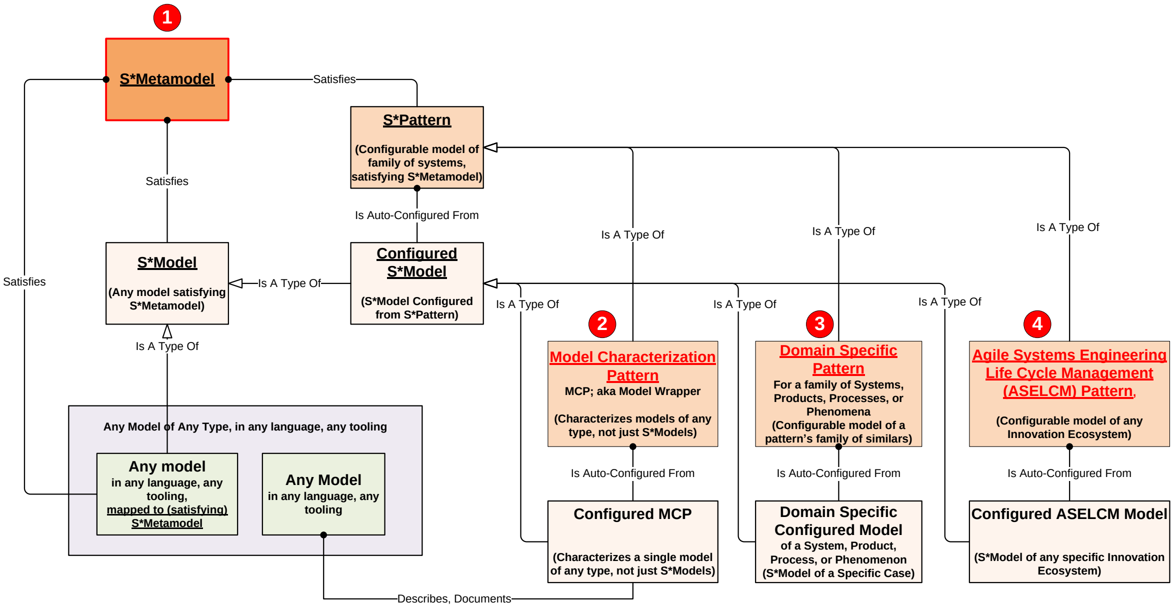


Figure 6: Model-Based Patterns Landscape

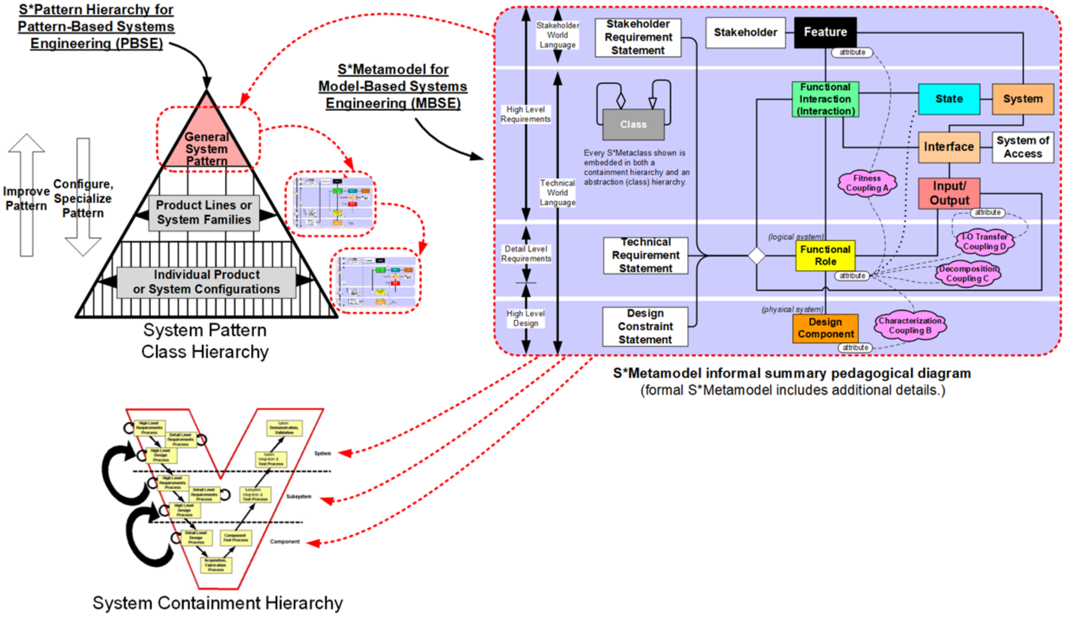
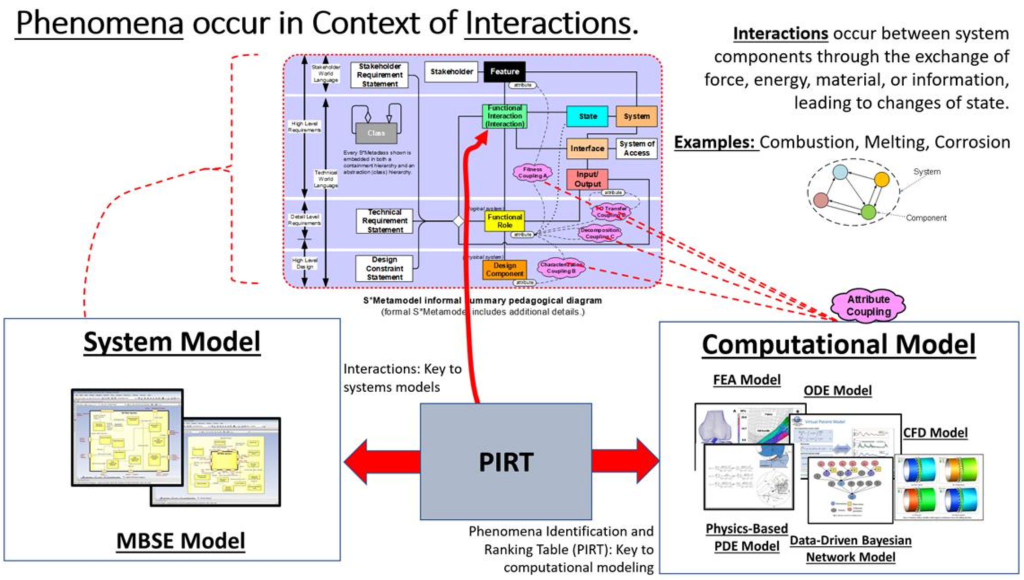


Figure 7: Simplified View of S*Metamodel as Modeling Framework for S*Models and S*Patterns


Phenomena occur in Context of Interactions.

Interactions occur between system components through the exchange of force, energy, material, or information, leading to changes of state.

Examples: Combustion, Melting, Corrosion

Interactions: Key to systems models

Phenomena Identification and Ranking Table (PIRT): Key to computational modeling

Figure 8: Formally Modeled Interactions Are Central to the S*Metamodel and to the Connection of SE Models to Computational Models, via a Phenomena Identification and Ranking Table (PIRT) (from [29])

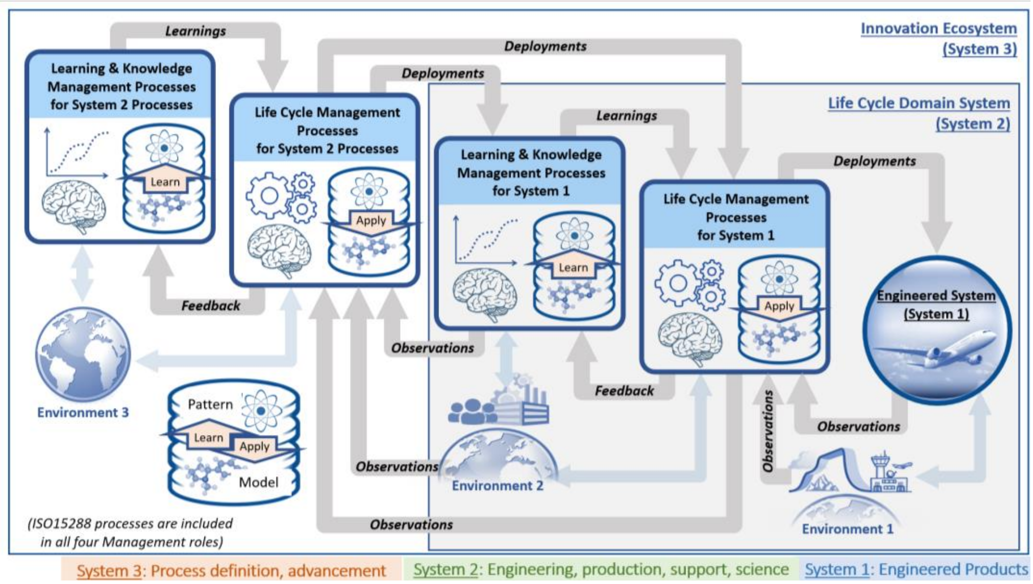


Figure 9: ASELCM Pattern, Level 1 Summary
(from AIAA Digital Thread Reference Model, [13])

Model-based patterns are deeply connected to the mathematical foundations of engineering practices of Section IIA above. That material recalls how the defined systems engineering discipline calls on practitioners to discover and validate a diverse set of information artifacts supporting the engineered product life cycle. However, that established framework of information consuming and producing processes is relatively silent on this simple question: "What about what we *already* know?" It is as if a project were finding and validating all that information for the first time – but that is virtually never the case; indeed, enterprise practitioners typically are well-acquainted with related information from past experience. However, the traditional ISO 15288 framework's relative silence on this question leaves it to practitioners to sort out the practical means of utilizing existing knowledge. This is part of the motivation for model-based patterns.

Since Pattern-Based Systems Engineering directly addresses making use of historical patterns of known information, it is sometimes suggested that this practice is mainly about what has come to be referred to as "reuse" of information. While information reuse is one key to pattern-based methods and highly valuable, there are deeper connections to be aware of between pattern-based methods and other established and powerful frameworks:

1. **Mathematics of Bayesian Methods**: When seeking and using new information, the very question "what about what I already know?" should ring familiar to anyone aware of the mathematics of Bayesian methods of conditional probability and related inference. Engineering has made dramatically successful use of these methods, such as Kalman filters [30], enabling deep space navigation and communication among other applications. Bayesian methods directly address the optimal mixing of what we already know with what we are just now finding out, in uncertain environments as a form of balanced learning—exactly our interest in CFD and other areas. A particularly valuable aspect of Bayesian models, such as Bayesian Networks, Bayesian Neural Networks, and Kalman Filters, is that they not only generate predicted responses and estimated states—they also generate estimated uncertainty levels of those outputs. Patterns encode what we already know—or could—and how certainly, as well as methodology. For a cautionary perspective on Bayesian methods in the sphere of model validation, see [11], Section 12.2.3.

2. **Mathematics of Experimental Design**: Accompanying the above question, we have its converse, "what about what I *don't* already know"? When we are trying to determine what underlying variables best characterize a phenomenon, how do know where to look for that which we don't even know yet? How do we avoid wasting time looking in the wrong places too long, when we don't know what right places even look like? These are the types of questions treated by the mathematics and methods of Design of Experiments (DOE) [31,32]. Model-based patterns encode what we are learning while we are still learning it, including uncertainties, as well as

methodology.

3. **Mathematics of Machine Learning (ML)**: After 75 years of progress in algorithms, hardware, data access, and mathematical understanding, machine learning has more recently attracted greater popular attention because of its dramatic demonstrations of progress. Model-based patterns encode what is learned by ML systems—or could be learned, as well as the algorithms of ML. Not all machine learning results in intuition-lacking network representations—ML systems also demonstrate the ability to learn Hamiltonians from observation. [33,34]. ML systems generate model hypotheses as well as measures of their validation. The mathematics of experiment design and learning have also come closer together, as in [35,36].

4. **Mathematics of Uncertainty Quantification and Model Validation**: CFD model validation is of particular interest in this paper. The large body of work by pioneers of computational model validation in general and CFD in particular, reviewed in part in [12], also contributes to what can be expressed in model-based patterns of both engineered systems and the processes of engineering. A perspective of the ways in which current patterns may need to change in order to address contemporary needs is provided in [7].

Patterns can be discovered by both human-performed research-based modelling as well as machine learning (ML). An attractive candidate for ML is the packaged characterization of computational models with respect to both the parameters of the conceptual model as well as the realization parameters of the computational model implementation, such as mesh and algorithm parametrics, even in the absence of expensive empirical data acquisition [37]. Although UQ may be undertaken using empirical data that may be expensive or not yet available, it is also fundamentally associated with the behavior of the model itself. This can be characterized in the absence of empirical data, packaged as a learned pattern, and distributed to downstream or future users who may later have access to empirical data. As described in [7], we are interested in validation over domains, not just individual points. Some of these domains may involve significantly different behaviors, phases, phenomena, or uncertainties. Configurability of pattern-based conceptual models may be exploited to represent these variabilities for downstream model users, as shown in Figure 10.

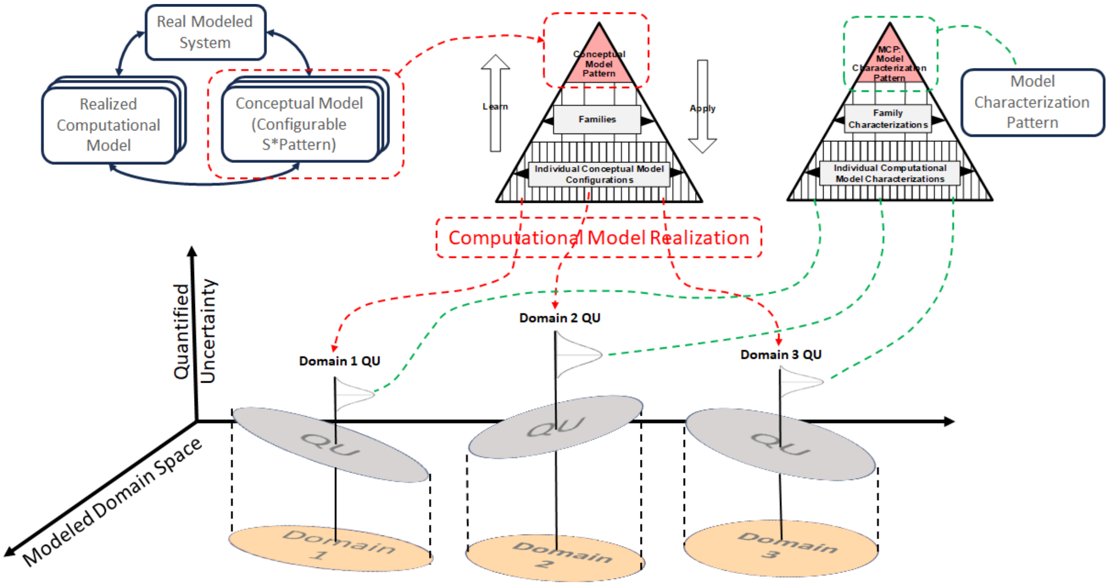


Figure 10: Conceptual Models Can Be Pattern-Based, Configured for Different Domains

III. Some Implications for CFD Validation

Having reviewed the essential principles underpinning the practice of Engineering Discipline, we now seek to show how they may be applied to the context of interest of this paper: CFD validation and, in particular, the shortfall in current capability compared with the NASA CFD 2030 Vision, as set-out in Section I. In this Section, we illustrate some aspects of the contemporary situation in the aerospace sector that have a bearing on the current state-of-the-art of CFD validation. The analysis presented relies heavily on that reported in [1]. Opportunities for using mutual accountability and validation dialog to help accelerate progress toward realizing this aspect of the NASA CFD 2030 Vision are presented in Section IV.

A. Multiple Perspectives; Multiple Lifecycles

[1] describes contemporary CFD model development lifecycles as being “loosely federated”. This means that the models being used in CFD workflows are generally developed within different lifecycles to those in which they are ultimately applied. Consequently, any limitations to the scope of the model validation completed prior to hand-over from one lifecycle to another may impose important constraints on its onward use. As [1] makes clear, there can be good pragmatic reasons for making a CFD model available for wider evaluation while local validation experience is extremely limited. However, the limited scope of the current CFD validation literature (see e.g. [3]) means that there are no published guidelines explaining how to consolidate model validation experience accumulated at isolated points into the domain-based evidence sought by most CFD users. As [1] also explains, there are many other (additional) factors and interactions that may influence the nature of the federation currently exhibited within CFD model development lifecycles.

In order to better understand current circumstances, it is helpful to recognise the diversity of interests on behalf of the various stakeholders involved. Simplifying things in order to convey some important aspects of the dynamics of the scenario, following [1], we identify two key dimensions within this stakeholder community: (i) the stage at which the actors are operating within the CFD model development lifecycle and (ii) the lifecycle to which stakeholders are contributing.

Regarding (i), [1] makes a simple distinction between Model Developers (MD) and Model Users (MU). Here, MD are the actors involved in whatever development activities are performed (formulation, implementation, calibration, validation, etc.) prior to the model being released for wider use. MU are the recipients of a fixed model.[*] Thus, the objective in redressing any deficit in the scope of the validation evidence supplied with the model (by the MD) is (rightly) confined to characterising the model behaviour only. Assessments regarding its suitability for a given task are promoted to a higher position in the MU lifecycle “Vee” – see Section II.A. This refinement to the objective of model validation in the context of a MU perspective is subtle, but of crucial importance. As stated at the top of this paper: model users require evidence of demonstrated model behaviour over a suitably wide range of model inputs upfront, so as to assess model suitability for a particular tasking.

Regarding (ii), [1] makes a distinction between model and aircraft development lifecycles. The latter can be incredibly complex (involving multiple customers, suppliers, etc. dispersed over huge – often intercontinental – distances …) and may extend over decades. However, despite the multitude, diversity and complexity of the factors that may influence their development, aircraft development lifecycles are managed as such. This situation contrasts dramatically with the model development lifecycle, for which, typically, there is no overall management or control. Here, for instance, the MD community (from the researchers devising new model formulations through to the organizations supplying configuration-controlled software) is naturally more fragmented. This observation is not intended as a criticism:[†] the diversity within the MD community is, in many ways, also its strength. However, coordination of these stakeholder interests presents a number of additional challenges to those faced in managing an aircraft development lifecycle. Put differently, while the completion of the model development “Vee” may be

---

[*] Note that modifications may be made to the model subsequently (for whatever reason – MD develop an improvement; MU express dissatisfaction, with supporting insights …). However, for the purpose of this simplified analysis, such modifications are considered MD rather than MU activities.

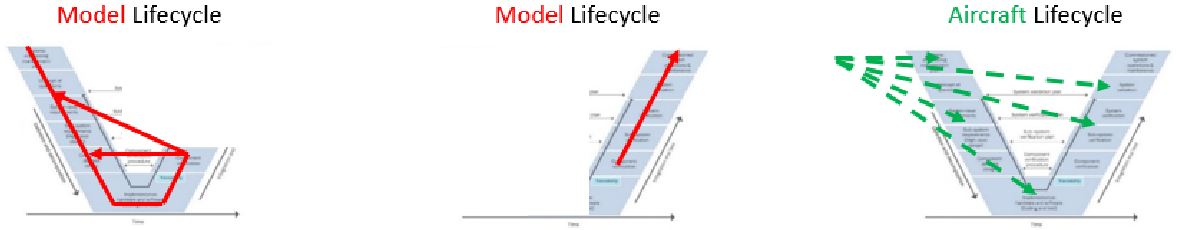
[†] The MU community is similarly dispersed – between different aircraft development lifecycles. Various factors (e.g. political and commercial) may inhibit the depth and frequency of the communication between them, even within a single industrial organization.

recognised as being mutually beneficial to all stakeholders, in the absence of an overarching development contract, the bonds of mutual accountability will need to be established in another way.

B. Diversity of Stakeholder Requirements and the NASA CFD2030 Vision

An inevitable consequence of the diversity within the stakeholder community is an accompanying diversity in the requirements associated with CFD model validation. Here we focus only on those which have been inferred from the quotation from the NASA CFD2030 Vision cited in Section I. The essential ingredients of these may be expressed as follows: succinctly stated, a validated CFD capability is required to be: (i) domain (not point) based, (ii) extrapolative, and (iii) available now (or at least before 2030). Further details are provided in [7].

Building on the analysis presented in Section III.A, one way of illustrating the current shortfall regarding CFD validation in the context of the NASA CFD2030 Vision schematically is presented in Figure 11. Here activities undertaken within the development cycle for a particular model are depicted in red; those contributing to an aircraft development lifecycle are shown in green.



(a) Model Developers (b) "Capability" Developers (c) Model Users

Figure 11: A schematic illustration of the fragmentation within contemporary CFD model development lifecycles

The schematic at left (Figure 11a), depicts the scope of activity undertaken by the MD community prior to the model being made available for wider use. The red lines and arrows may be interpreted as representing, for instance, scenarios associated with the development of a turbulence model – specifically its calibration (horizontal arrow, from right-to-left) or re-formulation (diagonal arrow pointing to an earlier stage in the lifecycle).[*]

The schematic at right (Figure 11c) depicts the potential scope of activity that may be undertaken by industrial aerodynamicists seeking to apply the model to whatever stage in the aircraft development lifecycle is currently "demanding" it. The green lines are dotted since, in the absence of validation evidence, significant effort may be required to re-dress this shortfall and establish the traceability required by Engineering Discipline.[†]

It is this significant effort that is depicted in the central schematic (Figure 11b). The associated "capability" development is depicted as belonging to the model development lifecycle (the activity arrow is coloured red). However, if the corresponding validation effort is not re-used elsewhere, it may be more appropriate to consider it as contributing to a specific aircraft development lifecycle since the work undertaken is likely to be highly context sensitive.

---

[*] For examples of these, we may consider the re-calibration of the production limiter in the Menter SST turbulence model [38], or its re-formulation to incorporate rotational and curvature corrections[39]. [40] provides a recent example of a turbulence model being made available for wider evaluation. (However, note that in this case, the awareness of – and respect for – caveats regarding its potential suitability are heightened by the novelty of the model formulation. Moreover, the scope of its inclusion into the turbulence modelling options provided by contemporary CFD codes is currently limited.)

[†] It should be noted that, at any one point in time, similar activities may be happening in multiple aircraft development lifecycles. In view of the ubiquity of some of the more established turbulence models, this situation extends to other types of engineered product in multiple sectors (aerospace, marine …).

In the context of this paper, the "capability development" activities of Figure 11b highlight the shortfall in the scope of contemporary CFD validation literature (see e.g. [3]). Certainly, for application to the left-hand leg of the aircraft development "Vee", the associated CFD validation requirements will be domain-based, extrapolative and sought immediately. It is difficult to imagine how continuing to confine CFD validation practice to "point-based" assessments will allow current the shortfall in capability to be addressed. So, how might mutual accountability be realised throughout the stakeholder community and in what ways can validation dialog be expected to accelerate progress toward realizing the NASA CFD2030 Vision?

IV.Mutual Accountability and Validation Dialog

There are many ways in which the authors believe that fostering improved mutual accountability and validation dialog will aid progress toward realizing the NASA CFD2030 Vision. For instance, as noted in Section II.A, both mutual accountability and validation dialog require communication between stakeholders. Clearly, for this to be effective, they need to understand one another. An important enabler for this is a shared clarity of purpose and expectation regarding their individual contributions in the wider enterprise (whatever that may be). Thus, benefits are to be expected simply by establishing a basic level of awareness concerning the practice of Engineering Discipline in general and by providing further detail regarding particular roles within it. Thus, further elaboration of analyses like that outlined in Section III, covering a wider and more extensive range of scenarios, in conjunction with material derived from that presented in Section II.A, will likely be helpful to both professional engineers and those who do not have a formal engineering background.

In this Section, we focus on those aspects of mutual accountability and validation dialog that are more specifically associated with validation, with a particular emphasis on explaining their implications for CFD model validation. To do so, we must first take a more fundamental look at what validation involves.

A. The essence of Model Validation

Within the framework of Engineering Discipline, the objective of model validation is *"the confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled"* [8]. There are multiple forms of dialog in play here. At its core, there is ultimately a mathematical dialog between the model being validated and the independent referent data being employed to facilitate the assessment. This also involves a dialog between the physical and digital (computational) realms – the former keeping the latter "honest" (see [41]). In addition to the mathematical operations involved, there is the (human) dialog between stakeholders that occurs before, during and after the maths has done its job. This is of crucial importance and serves a number of purposes in addition to the facilitation of mutual understanding described above. These include establishing the supporting reasoning (a key tenet in the practice of Engineering Discipline – see Section II.A) underlying the validation activity and providing the stimulus for identifying means for achieving improved outcomes. To provide further insight into how both of these latter purposes may be realised, we consider another fundamental consequence of model validation: that of the transition from assumptions to knowledge.

This transition is a fundamental to the development of all models (physical, computational …). When subjected to Engineering Discipline, the course of this transition is planned and managed in accordance with the principles set-out in Section II.A. Thus, for a computational model (including those employed in CFD), its origins are encapsulated in a series of assumptions (however derived – from observation through speculation) about the subject being modelled (turbulence, for instance). These assumptions may then be expressed in the form of a conceptual model, prior to being implemented in software and, subsequently, executed on hardware. Knowledge about the behaviour of the model, including (but not limited to) its accuracy, is then accumulated, progressively, by exercising the model on multiple occasions and in various computational environments. The option may be taken to revise the underlying assumptions (and corresponding aspects of the model) at any time during this process. However, ultimately, the knowledge accumulated is captured and quantified via model validation in accordance with the principles represented in Figure 12. Note the distinction between model accuracy assessment and model validation assessment. The former is consistent with "restricted" views of model validation (see e.g. [11]); the latter represents the progressive accumulation of knowledge and constitutes a method for achieving process closure when establishing a validated domain.[1][*]

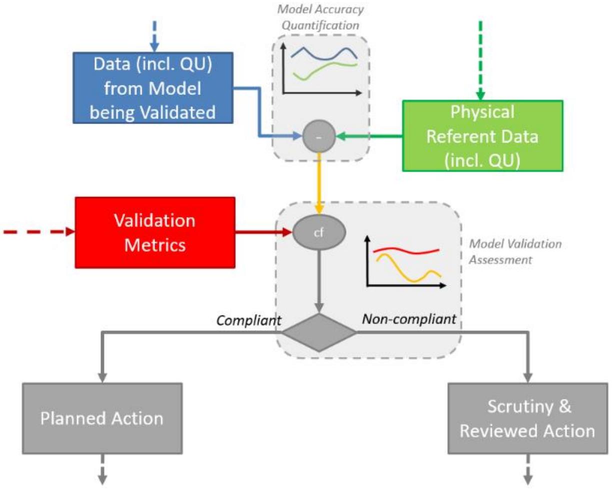


Figure 12: Schematic Stencil of a Single Model Validation Activity Post Preparation of Model & Referent Data and Updating of Validation Metrics (from [1]).
Note that only the aspect dedicated to model accuracy is depicted

As noted in Section II.A, the ISO standards refer specifically to two types of (computational model) validation: that utilised on the downward leg of the "Vee" (Requirements Validation); and that used on the upward leg (Solution Validation). Generally speaking, while the former may potentially include heuristic elements, the latter ought not to. However, this is not always the case. The choices made here are negotiated via stakeholder dialog and reflect their circumstances and constraints. Thus, the nature and maturity of the information secured by model validation may be expected to evolve during the course of the overarching lifecycle.[†] Model validation activities are planned accordingly.

To illustrate some of the factors considered when planning model validation activities, we introduce the concept of a Model Validation Landscape. This allows the way in which the dialog between the modelling assumptions and the physical behaviour being modelled – which lies at the heart of model validation – evolves to be monitored throughout the model development lifecycle.

B. Model Validation Landscape

A schematic of a model validation landscape is presented in Figure 13. A two-dimensional projection onto the landscape is depicted. Salient features of a third dimension, orthogonal to the plane shown, that convey supporting rationale (below the plane) and the validation metrics (above it) are described in [7].

---

[*] The concept of process closure in the context of model validation provides another example of how the objectives of a process may evolve as one passes through the lifecycle. For instance, during the model design stage, model calibration refers to the process of adjusting the model coefficients to better match measured physical data (a MD perspective). Later on once the design of the model has been fixed, calibration refers to the process of updating the validation metrics (QU) in accordance with the accumulated experience of exercising the model (a MU perspective).

[†] When subjected to scrutiny and analysis, the dynamics associated with the managed evolution of consistency threads, including those focussed on model accuracy, can be highly informative. Further comment on this subject is beyond the scope of this paper.

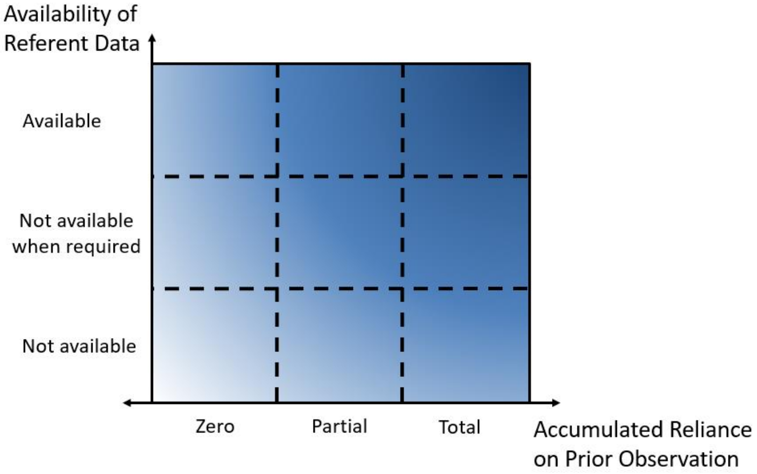


Figure 13: A Model Validation Landscape for a domain of interest

Figure 13 allows distinction to be made between the approaches that may be taken towards model validation (vertical axis) and the accumulated nature of the basis underpinning the current modelling assumptions (horizontal axis). In the interests of simplicity, three categories have been identified in each of these dimensions.* To explain how this works, we consider each of these axes in turn, below.

1. Vertical Axis

There is more than one approach that may be taken to effect model validation. Three distinct categories of approach are identified in Figure 14.

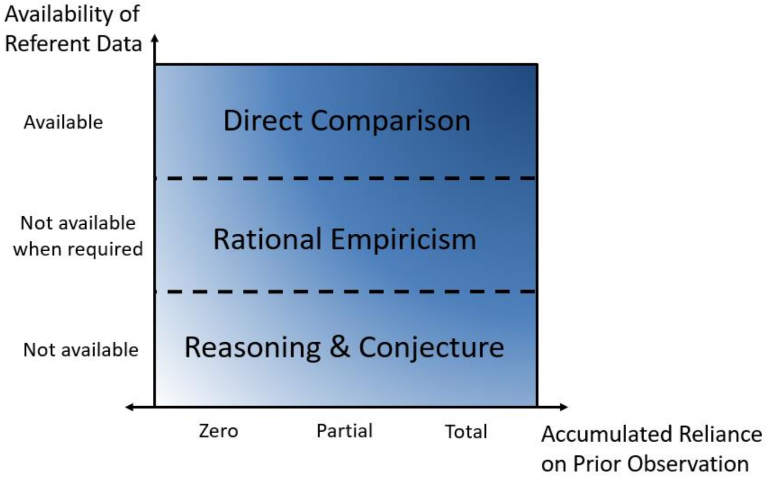


Figure 14: Categories of approach that may be taken towards model validation

---

* The proportions of the nine regions identified in Figure 13 are not drawn to scale. For the purposes of this paper, it is the existence of each region that is of primary concern. The precise nature of their demarcation (e.g. whether this be continuous, discrete or even overlapping, or whether further demarcation is beneficial) is of secondary importance. We believe the landscape is generic (i.e. it can be applied to any model): the reasons for drawing attention to the regions identified as "not available when required" on the vertical axis and "partial" on the horizontal axis will be explained below.

Direct Comparison refers to model validation workflows similar to that illustrated in Figure 12, in which the behaviour of the model being validated is compared directly with the independent data being used as a point of reference. Figure 12 depicts a model accuracy assessment and comparison with physical reference data; similar comments apply to any other quality of the model being validated (e.g. compatibility, reliability, usability …, see [10]), noting that is some cases the referent data may simply be a stated requirement (e.g. execution time, or cost) rather than a physical measurement. The subsequent model validation assessment pertains to the horizontal axis in the model validation landscape; as such, it is discussed further in Section IV.B.2, below.

Rational Empiricism typically involves more complex model validation workflows than that illustrated in Figure 12, since, by definition, they pertain to scenarios for which referent data is not available at the time of the planned validation activity. This additional complexity primarily affects only the model quality (accuracy) assessment: the mechanisms for effecting process closure (i.e. model validation assessments) are largely unaffected. Rational empiricism may be applied during the establishment of interpolative and extrapolative validation domains. Further details are provided in [7].

In circumstances where no (or extremely limited) referent data is available, it may be necessary to rely upon reasoning and conjecture. As noted above, such heuristic assessments are not uncommon in the context of Requirements Validation – where they are increasingly being referred to as pertaining to Model Suitability (see e.g. [42]). There may also be circumstances where this type of approach is appropriate for Solution Validation. The reliance on reasoning and conjecture in model validation is an important topic since it overlaps with the concept of Model Credibility. Accordingly, this topic is re-visited in Section IV.C.2, below.

2. *Horizontal Axis*

Following a similar rationale to that associated with the vertical axis, there is more than one approach that may be adopted to derive assumptions about the subject to be modelled. Again, for convenience and simplicity, the horizontal axis of the model validation landscape (Figure 13) has been segmented into three regions in an attempt to straddle the options available. For example, one can appeal to established physical laws or focussed observations. (In the context of CFD turbulence models, the assumptions concerning the behaviour of boundary layers on flat plates fall into this category). These fall within region of the horizontal axis labelled "Total". Alternatively, one may rely – at least in part – on more speculative reasoning, in the manner of Milton Friedman.[43] (Certain assumptions embedded in contemporary turbulence models also fall into this category). Such scenarios fall broadly within the region of the horizontal axis labelled "Partial". Additionally, in the absence of any clear pattern in past observations, one may apply machine learning techniques to identify patterns that had previously defied the human eye. (Turbulence modelling has also been subjected to this type of analysis.) This situation would be appropriate to the region labelled "Zero".

Unlike the vertical axis in the model validation landscape, the horizontal axis is also intended to reflect the progressive accumulation of assumptions over-and-above those embedded in the model itself. This is closely related to the maturation of the model validation metrics used in model validation assessments (see e.g. Figure 12). Increases in maturity correspond to a translation to the right on the horizontal axis of Figure 13. Thus, even if the original formulation was considered to merit placing it somewhere in the "Partial" region on the horizontal axis, the maturation of the validation metrics would eventually allow it to transition into the "Total" region. Similarly, if a model previously considered to be totally reliant on prior observations is to be validated outside its original (or a previously proven) context of use – i.e. for model inputs that have not been assessed previously – this would result in a transition to the left, potentially moving the current validation status from "Total" to "Partial". Note that the basis for establishing the current location on the horizontal axis of the model validation landscape is essentially heuristic (i.e. judgement-based, evaluated in terms that are meaningful to their context). Quantified information, such as that defining the characteristics the domain of interest (range of model inputs addressed, for instance) and all related QU (together with its support) resides in a plane orthogonal to that illustrated in Figures 13 and 14: further details of this are provided in [7].

3. *Anticipated patterns of use*

Having explained how the model validation landscape may be used to capture, monitor and update the evolving nature of the demonstrated relationship between the modelling assumptions and the physics being modelled, it is now possible to consider how this facility may be used. There are many potential possibilities. However, for the purposes of this paper, we focus our discussion on its use to support the planning of model validation activities in accordance with the principles of Engineering Discipline.

When planning the model validation to be undertaken within an engineering lifecycle, an understating of the overarching requirements, including details of the ways in which their achievement is to be measured and demonstrated is essential. The principles of the approach to be adopted are agreed between the relevant stakeholders and a validation plan established. Associated risks are also identified and a management plan established to ensure these are managed throughout the lifecycle.

Throughout, it is crucial to understand how a particular model validation activity contributes to realizing the overarching model validation requirements. In the Solution Validation phase, this involves ensuring that the current status of the demonstrated relationship between the modelling assumptions and the physics being modelled is understood and, equally importantly, that participants have a shared appreciation of the basis by which this relationship is expected to be updated in accordance with what is learned during the validation activity. This has a strong bearing on the choice of validation approach to be adopted (vertical axis of the model validation landscape).

For instance, for a limited-scale blue-sky research activity (a PhD research project, for instance), the potential for undertaking comparisons with physical data may be limited in scope and, as a result, there may be a strong reliance on Reasoning and Conjecture (Figure 14). Neither of these constraints need constitute a barrier to successful completion of the engineering lifecycle or the award of a PhD. If the computational model is intended primarily to support process control or problem diagnosis applications, a validation approach focussed on Direct Comparison (Figure 14) may be appropriate. However, if the intent is to support a design capability, then there is likely to be a reliance on Rational Empiricism (Figure 14) in model validation to establish extrapolative validated domains of the requisite scope. It should be stressed that the above examples are only illustrative: they are not conceived as being prescriptive or definitive.

C. Implications

Of the many potential implications arising from the above analyses, we address only two: (i) the search for patterns in fluid flow data and (ii) the relationship with model credibility.

1. The search for patterns in fluid flow data

As explained in [7], one of the factors contributing to the contemporary shortfall in capability with regard to the requirements of the NASA CFD 2030 Vision (as outlined in Section I) is that the uncertainty quantification (UQ) processes being developed within the UQ community are primarily point-based. The development of validated, domain-based, extrapolative CFD capabilities will require alternative approaches to the quantification of uncertainty to be developed. Such approaches will need to address multiple challenges, not least the establishment of effective mechanisms for demonstrating validation process closure over potentially extensive ranges of model inputs.

A subsidiary aspect of this is associated with the topic of modelling scenario complexity – be this associated with the complexity of the physics being simulated or the complexity of the boundary conditions being imposed, or both. In other words, how does one progress from simplified, idealised scenarios that may form the basis of initial study and model formulation, through to circumstances of more practical industrial interest (the airflow about a fully-configured airframe, for instance)?

A number of heuristic techniques have been developed to help stimulate progress in this regard. One of the more widely known is the Model Validation Hierarchy (MVH)[11]. While MVH may have proved helpful in other fields of scientific or engineering endeavour, there is little published evidence demonstrating their use and utility in support of the development of CFD. In a nutshell, the unresolved questions revolve about the basis for ascendancy through the MVH: current approaches are heuristic in nature and not subject to formal deterministic analysis.[44] [*] [†]

---

[*] [45] provides some valuable insights into the underlying principles and limitations.

[†] For instance, the principal author has yet to see an MVH for CFD that relies, in any way, upon a mapping to the underlying model assumptions (which are essentially constitutive – not phenomenological – in nature). There would therefore appear to be open questions concerning the correspondence between the constitutive nature of the modelling assumptions and their correlation with the behaviours that lead to the formation of observable phenomena (like vortices, shock waves or shear layers). Are there any patterns to be observed here that might form a basis for ascending a MVH? Put simply, should we be seeking a “lowerarchy” in order to develop a hierarchy?

How we should seek to abstract such problems and whether we should approach them from a bottom-up or top-down direction – or maybe a combination of both – all seem to be open questions. However, the transition from point-based to domain-based processes for establishing QU envisaged herein may provide both (i) a renewed stimulus for seeking such patterns out and (ii) multiple opportunities for finding them. Should such patterns be found, they may be of benefit to both MD and MU communities. For instance, improved insights into any patterns governing constitutive behaviours of the fluid may lead to improved formulations of turbulence models. Similarly, improved insights into the patterns of disparity between measured and computed fluid flow behaviour may be expected to guide the formulation of improved model validation metrics. In view of the fragmented (loosely federated) nature of CFD model development lifecycles (Section III.A), could it be that prospects like these may help foster bonds of mutual accountability throughout the stakeholder community?

2. The relationship with model credibility

A primary concern in the practice of Engineering Discipline is that considerations pertaining to credibility (in all its forms) are addressed directly within the engineering lifecycle by the stakeholders developing an agreed approach to Solution Validation up-front. Progress against this is monitored – and plans updated as required – throughout. For the development of commercially available products (including computational models), Solution Validation would be expected to have little or no reliance on unvalidated heuristic measures. However, for development lifecycles that are subject to recognised constraints regarding the availability of objective referent data (e.g. associated with the risk, cost and/or feasibility of securing it), this expectation may be relaxed, by mutual consent. As a result, the approach taken towards model validation in these circumstances may appear further left in the model validation landscape than would otherwise be expected. There are many possibilities here. Moreover, there would appear to be strong overlap with the concept of model credibility in these circumstances.

Thus, since "model credibility" is grounded in point-based approaches to model validation, the pursuit of domain-based, extrapolative approaches to CFD validation may create opportunities to gain a more profound and balanced view of the developments that have been made in UQ within the MD community. This is another topic that would benefit from fostering mutual accountability and validation dialog throughout the stakeholder community.

V. Closing Remarks

Within a framework of Engineering Discipline, the essence of validation is *"the confirmation, through the provision of objective evidence, that the requirements for a specific intended use or application have been fulfilled"*. By appealing to principles that have been refined and proven over many decades throughout the engineering and scientific communities, this paper has shown how the practices that have been developed specifically to support computational model validation may be viewed consistently with other means of accomplishing (requirement and solution) validation.

When considered within the context of an overarching engineering lifecycle, one of the principal functions of computational model validation is to record the progressive accumulation of knowledge regarding the effectiveness of the assumptions being made about the behaviour being modelled. Introducing the concept of a model validation landscape has enabled the various approaches that may be taken toward computational model validation – from the heuristic to the rigorously quantified – to be viewed from a consistent perspective.

The overarching engineering lifecycle of the computational models used in CFD are rarely facilitated by a single (or a coordinated suite of) development contract(s). Consequently, the mutual accountability between stakeholders evident in the development of many other types of engineered product is harder to attain. The diversity within the computational model development community is both a strength and, if judged by the lack of progress made towards realising the aspects of the NASA CFD2030 Vision described at the top of this paper, also an area of weakness. If new ways of fostering mutual accountability between stakeholders can be established, we can expect elevated levels of performance to result. Therefore, the authors propose refining the analysis and principles described herein with a view toward scrutinising them in the context of multiple, diverse computational model development scenarios. The scope of this endeavour should extend beyond CFD and beyond the aerospace sector. One of its more immediate goals should be the publication of a broader series of guidelines on computational model validation that serve the stakeholder community as a whole.

References

[1] Taylor, N.J., “CFD Validation: What is it? & How do we effect it?”, AIAA 2022-3768, June 2022.

[2] Slotnick, J., et al., “CFD Vision 2030 Study: A Path to Revolutionary Aerosciences,” NASA/CR-2014-218178, 2014.

[3] “Standard for Verification and Validation in Computational Fluid Dynamics and Heat Transfer”, ASME V&V 20-2009 (R2019), ASME, New York, 2019.

[4] Kraft, E.M., “Why Hasn’t the Computer Replaced with Wind Tunnel?”, ITEAJ 31: 329–346, 2010.

[5] Cary, A., et al., “CFD Vision 2030 Road Map: Progress and Perspectives”, AIAA 2021-2726, August 2021.

[6] Taylor N.J., “CFD Validation - A Perspective on the Current State of the Art”, paper to be presented at AIAA SciTech 2024.

[7] Taylor, N.J. and Romero, V., “CFD Validation: From Uncertainty Quantification to Quantified Uncertainty – Potential Paths toward Improved Feasibility and Reach”, paper to be presented at AIAA SciTech 2024.

[8] ISO, "ISO/IEC/IEEE 15288-2023: ISO/IEC/IEEE International Standard - Systems and software engineering -- System life cycle processes", 2023.

[9] Walden, D., et al, eds., “INCOSE Systems Engineering Handbook”, Fifth Edition, International Council on Systems Engineering, San Diego, Ca, 2023.

[10] ISO, ISO/IEC 25010:2023 “Systems and Software Engineering Systems and Software Quality Requirements and Evaluation” (SQuaRE), 2023.

[11] Oberkampf, W. and Roy, C., “Verification and Validation in Scientific Computing”, Cambridge U. Press, 2010.

[12] Oberkampf, W. and Trucano, T., 2002. "Verification and Validation in Computational Fluid Dynamics", Sandia SAND2002-0529, 2002.

[13] Cribb, M., et al., "Digital Thread: Definition, Value, and Reference Model", AIAA, 2023.

[14] French, M., et al, “Digital Twin: Reference Model, Realizations & Recommendations”, AIAA, 2023.

[15] Schindel, W., Sherey, J., Lewis, S., and Sanyal, S., "Accelerating MBSE Impacts Across the Enterprise: Model-Based S*Patterns", in *Proc of 2015 INCOSE International Symposium*, Seattle, Wa, 2015.

[16] Schindel, W., “Failure Analysis: Insights from Model-Based Systems Engineering”, in *Proc of INCOSE 2010 International Symposium*, July 2010.

[17] LeCun, Y., Bengio, Y., & Hinton, G., "Deep Learning", *Nature*, Vol 521, pp. 436-444, 2015.

[18] Stern, F, Diez, M., Sadat-Hosseini, Yoon, H, and Quadvlieg, F., "Statistical Approach for Computational Fluid Dynamics State-of-the-Art Assessment: N-Version Verification and Validation" in *ASME J. of Verification, Validation, and Uncertainty Quantification*, Sept 2017.

[19] Schlesinger, S., "Terminology for Model Credibility", *Simulation*, 32(3), 103-104, 1979.

[20] Schindel, W., "Got Phenomena? Science-based Disciplines for Emerging Systems Challenges”, *Proc of INCOSE 2016 International Symposium*, 26(1), 2256–2271, 2016.

[21] Schindel, W., "SE Foundation Elements: Discussion Inputs to INCOSE Vision 2035 Theoretical Foundations Section”,V2.3.2a., INCOSE Patterns Working Group. 2020. Download from https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:science_math_foundations_for_systems_and_systems_engineering--1_hr_awareness_v2.3.2a.pdf

[22] Schindel, W., "Pattern-Based Methods and MBSE", chapter 7 of *Handbook of Model-Based Systems Engineering* , Augustine, N., and Madni, A, eds., 2023. Download from https://link.springer.com/referenceworkentry/10.1007/978-3-030-27486-3_73-1

[23] Hightower, et al, "Verification and Validation Interactions with the Model Life Cycle: Status of a VV50 Working Group", *Proc of ASME V&V Symposium*, May 2021.

[24] Schindel, W., "What Is the Smallest Model of a System?”, *Proc of INCOSE International Symposium*, 2011. Download from -- https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:what_is_the_smallest_model_of_a_system_v1.4.4.pdf

[25] INCOSE Patterns Working Group, “Systematica® Metamodel”, Metamodel Version 8.0, Document Version 8.0.34, April 7, 2022. Download from https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:systematica_6_metamodel_v8.0.34.pdf

[26] INCOSE Patterns Working Group, "The Model Characterization Pattern (MCP): A Universal Characterization & Labeling S*Pattern for All Computational Models", V1.9.3, 2019. Download from https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:model_characterization_pattern_mcp_v1.9.3.pdf

[27] Paredis, Sirin, Yannou, Coatanea, and Landel, "A Model Identity Card to Support Simulation Model Development Process in a Collaborative Multidisciplinary Design Environment", IEEE Systems Journal, 8 January 2015.

[28] INCOSE Patterns Working Group, "Methodology Summary: Pattern-Based Systems Engineering (PBSE), Based On S*MBSE Models", 2019. Download from https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:pbse_extension_of_mbse--methodology_summary_v1.6.1.pdf

[29] INCOSE Patterns Working Group, "Consistency Management as an Integrating Paradigm for Digital Life Cycle Management with Learning", 2021. Download from— https://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:patterns:aselcm_pattern_--_consistency_management_as_a_digital_life_cycle_management_paradigm_v1.3.1.pdf

[30] Kalman, R., "A New Approach to Linear Filtering and Prediction Problems," Transactions of the ASME–Journal of Basic Engineering, Vol. 82-D, pp. 35–45, 1960.

[31] Fisher, R., "The Design of Experiments", Hafner Press, 1935.

[32] Box, G., Hunter, J., and Hunter, W., "Statistics for Experimenters: Design, Innovation, and Discovery", 2nd Edition, Wiley, 2005.

[33] Toth, P., et al., "Hamiltonian Generative Networks", in Proc. of the 2020 International Conference on Learning Representations. Addis Ababa, Ethiopia, 2020.

[34] Greydanus, S., et al. "Hamiltonian Neural Networks", in Proc. of NeurIPS 2019.

[35] Mehta, V., et al., "An Experimental Design Perspective on Model-Based Reinforcement Learning", in Proc of International Conference on Learning Representations (ICLR), 2022.

[36] Neiswanger, W., Wang, K. and Ermon, S., "Bayesian Algorithm Execution: Estimating Computable Properties of Black-Box Functions Using Mutual Information", in Proc 38th Conf on Machine Learning, 2021.

[37] Matha, M. and Kucharczyk, K., "Applicability of Machine Learning in Uncertainty Quantification of Turbulence Models", 2022.

[38] Menter, F. R., Kuntz, M., and Langtry, R., "Ten Years of Industrial Experience with the SST Turbulence Model," Turbulence, Heat and Mass Transfer 4, pp. 625 - 632, 2003.

[39] Smirnov, P. E., Menter, F. R., "Sensitization of the SST Turbulence Model to Rotation and Curvature by Applying the Spalart-Shur Correction Term," ASME Journal of Turbomachinery, Vol. 131, October 2009.

[40] Rumsey, C. L., et al., "Improvements to the Quadratic Constitutive Relation Based on NASA Juncture Flow Data," AIAA Journal, Vol. 58, No. 10, pp. 4374-4384, 2020.

[41] Taylor, N.J. and Rumsey, C.L., "CFD Validation Experiments: Toward A Broader Perspective", AIAA 2021-1933, January 2021.

[42] Chawner, J.R. and Taylor, N.J., "Progress in Geometry Modeling and Mesh Generation Toward the CFD Vision 2030", AIAA 2019-2945, June 2019.

[43] Schruben, L.W., "Establishing the credibility of simulations", Simulation v.34, pp.101–105, 1980.

[44] Taylor, N., et al., "A Process Framework for Identifying Requirements for Physical Referent Data to Support Computational Model Validation", AIAA 2023-2606, January 2023.

[45] Hills, R.G., "Roll-up of Validation Results to a Target Application", Technical Report, SAND 2013-7424, 2013.