SYSTEMS ENGINEERING
Research Center

Transforming System Engineering through Model-Centric Engineering

Technical Report SERC-2015-TR-044-3

January 31, 2015

Principal Investigator: Dr. Mark Blackburn, Stevens Institute of Technology

Research Team:

Dr. Rob Cloutier, Stevens Institute of Technology

Dr. Gary Witus, Wayne State University

Eirik Hole, Stevens Institute of Technology

Mary Bone, Stevens Institute of Technology

Sponsor:

NAVAIR, DASD (SE)

Contract Number: HQ0034-13-D-0004 **Task Order 0018, RT 118**

Report No. SERC-2015-044-3

Report Documentation Page

Form Approved
OMB No. 0704-0188

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
31 JAN 2015	**N/A**	

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Transforming System Engineering through Model-Centric Engineering	**HQ0034-13-D-0004**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) **Blackburn /Mark**	5d. PROJECT NUMBER **RT 118**
	5e. TASK NUMBER **TO 018**
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stevens Institute of Technology**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) **DASD (SE)**	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The vision held by NAVAIRâ\,s leadership is to assess technical feasibility of radical transformation through a more holistic model-centric engineering approach. The expected capability of such an approach would enable mission-based analysis and engineering that reduces the typical time by at least 25 percent from what is achieved today for large-scale air vehicle systems. Research tasks were: surveying industry, government and academia to understand the state-of the-art of a holistic, model-centric engineering, develop a common lexicon model-centric engineering, model the Airworthiness processes, and integrate a risk management framework. While research suggests that it is technically feasible to create a holistic approach for conceiving innovative concepts and solutions enabled through model-centricity. Operational transformation is needed to coordinate efforts across multiple disciplines and multiple stakeholders. Findings explored with NAVAIR are: growth, complexity, and development of software; interoperability, consistency, and limitations of transforming model semantics; and validation of models.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**122**	

Copyright © 2015 Stevens Institute of Technology

The Systems Engineering Research Center (SERC) is a federally funded University Affiliated Research Center managed by Stevens Institute of Technology.

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract HQ0034-13-D-0004 (TO 0122).

Any views, opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Department of Defense nor ASD(R&E).

No Warranty.

This Stevens Institute of Technology and Systems Engineering Research Center Material is furnished on an “as-is” basis. Stevens Institute of Technology makes no warranties of any kind, either expressed or implied, as to any matter including, but not limited to, warranty of fitness for purpose or merchantability, exclusivity, or results obtained from use of the material. Stevens Institute of Technology does not make any warranty of any kind with respect to freedom from patent, trademark, or copyright infringement.

This material has been approved for public release and unlimited distribution.

ACKNOWLEDGMENTS

We wish to acknowledge the great support of the NAVAIR sponsors and stakeholders, including stakeholders from other industry partners that have been very helpful and open about the challenges and opportunities of this promising approach to transform systems engineering.

We want to specifically thank Dave Cohen who established the vision for this project, and our NAVAIR team, Jaime Guerrero, Gary Strauss, and Ron Carlson, who has worked closely on a weekly basis in helping to collaboratively research this effort. Gary Strauss has taken on the deputy role for Eric (Tre´) Johnsen, who had taken on the deputy role for Larry Smith, and Ernest (Turk) Tavares, who we also thank for their support and contributions.

We also want to thank all, currently more than 180 stakeholders that participated in 29 organizational discussion and working session, including some from industry that will remain anonymous in recognition of our need to comply with proprietary and confidentiality agreements associated with Task 1.

As there are so many stakeholders that supported this effort, we wish to recognize them all. We sincerely apologize if we have missed anyone else that has supported our efforts.

Andrew Devine	Dale Moore	John Funk	Philomena Zimmerman
Art Pyster	Dennis Reed	John McKeown	Richard Price
Aumber Bhatti	Doreen Dixon	Joseph Tolarski	Richard Yates
Bill Brickner	Doris Schultz	Judith Dahmann	Scott Lucero
Bhawana Sharma	Fatma Dandashi	Leslie Taylor	Shahram Bavani
Brad Kidwell	Fran Chamberlain	Lissette Fortuno	Stan Rifkin
Brian Nolan	Howard Owens	Michael Gaydar	Stu Young
Brent Gordon	Jae Pfeffer	Michael Alcantara	Todd Standard
Bob Levitt	James Carroll	Mike Kelley	Tom Blakely
Chris Owen	Jeff Smallwork	Paul Montgomery	

TABLE OF CONTENTS

LIST OF FIGURES AND TABLES

EXECUTIVE SUMMARY

This is the final report for phase 1 of the Systems Engineering Research Center (SERC) research task (RT-118). The RT focuses on a Vision held by NAVAIR's leadership to assess the **technical feasibility** of **radical transformation** through a more holistic model-centric engineering approach. The expected capability of such an approach would enable mission-based analysis and engineering that reduces the typical time by at least 25 percent from what is achieved today for large-scale air vehicle systems. The research need includes the evaluation of emerging system design through computer (i.e., digital) models. The effort extends RT-48 to investigate the technical feasibility of moving to a "complete" model-centric lifecycle and includes four overarching and related tasks as shown in Figure 1. These tasks include:

- Task 1: Surveying Industry, Government and Academia to understand the state-of the-art of a holistic approach to model-centric engineering ("everything digital")
- Task 2: Develop a common lexicon for things related to models, including model types, levels, uses, representation, visualizations, etc.
- Task 3: Model the "Vision," but also relate it to the "As Is" and Airworthiness processes
- Task 4: Integrate a Risk Management framework with the Vision

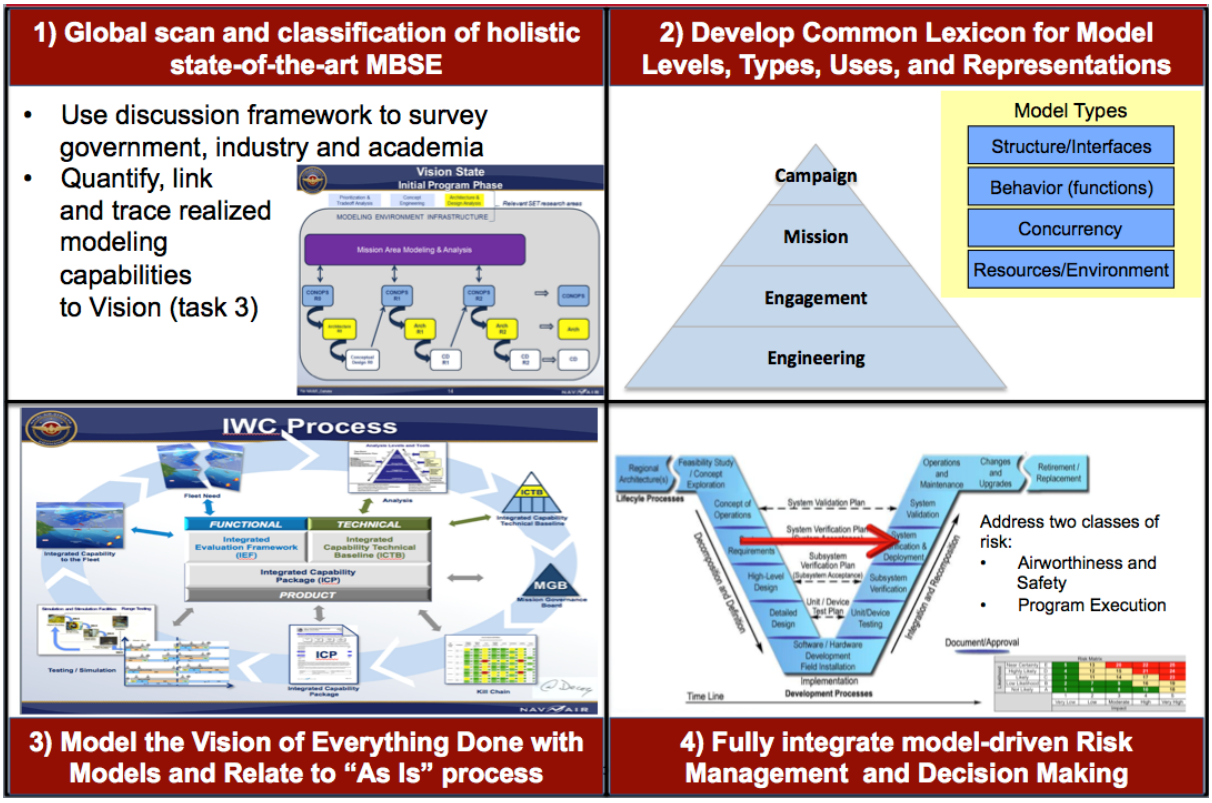


Figure 1. Four Tasks to Assess Technical Feasibility of "Doing Everything with Models"

Since the kickoff of RT-118 there has been considerable emphasis on understanding the state-of-the-art through discussions with industry, government and academia. We have conducted over 29 discussions, including 21 on site, as well as several follow-up discussions on some of the

identified challenge areas. We did not do a survey, but rather had open-ended discussions. We asked the meeting coordinators to in general:

> Tell us about the most advanced and holistic approach to model-centric engineering you use or have seen used.

The spectrum of information was very broad; there really is no good way to make a comparison. In addition, we had proprietary information agreements with most industry organizations. The objective was not to single out any specific organization, therefore, we will summarize, in the aggregate, what we heard in this report as it relates to the NAVAIR research objective.

Research Result

Our research suggests that model-centric engineering is in use and adoption seems to be accelerating. Model-centric engineering can be characterized as an overarching digital approach for integrating different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity across disciplines throughout the lifecycle. We seem to be getting closer to a tipping point and progressing beyond model-based to model-centric where integration of computational capabilities, models, software, hardware, platforms, and humans-in-the-loop allows us to assess system designs using dynamic models and surrogates to support continuous and often virtual verification and validation in the face of changing mission needs.

Enabling digital technologies are changing how organizations are conceptualizing, architecting, designing, developing, producing, and sustaining systems and systems of systems (SoS). Some use model-centric environments for customer engagements, as well as design engineering analyses and review sessions. While they do use commercial technologies, most have been innovating and have developed a significant amount of enabling technology – some call it their "secret sauce." We have seen demonstrations of mission-level simulations that are being integrated with system simulation, digital assets and aircraft products providing cloud-like services enabled by the industrial Internet. We have seen demonstrations of 1D, 2D, and 3D modeling and simulations with a wide array of solvers and visualization capabilities. We have been in an immersive Cave Automated Virtual Environment. We have seen the results of platform-based approaches directly focused on speed-to-market, and more, which is discussed in Section 2.

Model-centric engineering technologies enable more automation and efficiencies, however while research suggests that it is technically feasible to create a holistic approach for conceiving innovative concepts and solutions enabled through model-centricity, we need a radical transformation to change how we operate to coordinate the efforts across multiple disciplines with all relevant stakeholders at the right time and virtually. While there are still challenges, there are many opportunities.

We will discuss some challenge areas in Section 2.5, but a few examples are summarized here:

1. The discussions with organizations often stated known facts such as 90 percent of the functionality in a 5th generation air vehicle system is in software

- The growth and complexity of software requires a significant amount of software verification, which is essential to airworthiness and safety, but often resulting in longer than expected durations and schedule slips

- The aspects of software were not originally high on the list, but in model-centric engineering, software connects almost everything, and while the impact of software was not believed to be an issue in the past, it is likely to be going forward; this has been confirmed in many discussions and the implications to NAVAIR need deeper investigation

2. It was stated in meetings that there is an "explosion of models," however, there is a lack of cross-domain model interoperability, consistency, and limitations transforming models with the required semantic precision to provide accurate information for decision making

3. It was stated that unvalidated models are used leading to incorrect or invalid results leading to organizations not identifying design or integration problems until late in the lifecycle

This list is not comprehensive. This report provide some scenarios about how to address item #1, and we will have some follow-up discussions with organizations, and further investigate root causes, which might be addressed by early modeling and simulation to produce "better" requirements. We also think item #2 can be addressed through "engineering," and NAVAIR is making some headway on this item.

As for item #3, this topic relates to a question posed by our sponsor after our review of the material presented in this report, paraphrased:

> If we are going to rely more heavily on model-centric engineering, with an increasing use of modeling and simulations, how do we know that models/simulations used to assess "performance" have the needed "**integrity**" to ensure that the performance predictions are accurate (i.e., that we can trust the models)?

Our visit to Sandia National Laboratory and one industry organization provided some insights into model-centric approaches and tools they are using that can address this topic. We believe that their approach and tools provide a measure of certainty into a model's predictive capabilities, and measures of uncertainty of these predictive capabilities can apply to almost any model/simulation.

We believe our research finding address most aspects of the research questions. The report discusses implications and alternatives to the transformation for NAVAIR. We have had follow-ups to our meetings on several different topics, and have more planned that are focused on some of the challenge areas. We have been asked to bring industry together to share their perspectives on challenges, issues, concerns, and enablers for a transformation. The concept for a radical transformation still needs to be addressed as we move forward.

INTRODUCTION

In 2013, the Naval Air Systems Command (NAVAIR) at the Naval Air Station, Patuxent River, Maryland initiated a research task (RT-48) to assess the technical feasibility of creating/leveraging a more holistic Model-Based Systems Engineering (MBSE) approach to support mission-based analysis and engineering in order to achieve a 25 percent reduction in development time from that of the traditional large-scale air vehicle weapon systems. The research need was focused on the evaluation of emerging system design through computer (digital) models. The first phase of the effort created a strategy and began collecting and structuring evidence to assess the technical feasibility of moving to a "complete" model-driven lifecycle. The second phase conducted under RT-118 and documented in this report involved an extensive outreach to understand the state-of-the-art in using models. We conducted over 29 discussions, 21 on site with industry, government and academia, as well as several follow-up discussions.

A goal is to leverage virtual designs that integrate with existing systems data and simulations, as well as surrogates at varying levels of refinement and fidelity to support a more continuous approach to mission and systems analysis and design refinement. This broader view of the use of models has moved our team to use the term model-centric engineering. Model-centric engineering can be characterized as an overarching digital approach for integrating different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity, including software-, hardware-, platform-, and human-in-the-loop across disciplines throughout the lifecycle.

The larger context of the NAVAIR mission seeks a radical transformation through model-centric engineering. The Vision of NAVAIR is to establish an environment to evaluate the emerging system design through computer models and demonstrate system compliance to user performance and design integrity requirements, while managing airworthiness risks. It is anticipated that this model-centric approach can streamline and radically transform the traditional document-centric process that decomposes requirements and their subsequent integrated analysis, which is currently aligned with the Department of Defense (DoD) systems engineering V-model (i.e., the "V"). By providing more tightly coupled and dynamic linkages between the two sides of the traditional "V," more efficient and focused requirements decomposition would eliminate thousands of pages of documentation delivered via contract data requirements that now substitute for directly invoking, manipulating, and examining the design through computer-based models.

OBJECTIVE

This transformation initiative for NAVAIR is broad and can be thought about in at least three parts as it relates to our task:

1. The focus of this research task, RT-118, is scoped at the system level, sometimes characterized as the Program of Record (POR) plus weapons, for an air vehicle system

2. There is another related effort focused at the mission level, involving systems of systems (SoS), which must consider capabilities cutting across platforms of systems
3. There is a third effort focused on transitioning through adoption of model-centric engineering, with focus on adopting "agile" practices; while this effort may leverage model-centric engineering technologies that improve automation and efficiencies, it is not necessarily "radically transformative"

While our directive is to focus on the technical feasibility of a radical transformation for item #1, our discussions with organizations and working sessions involving other stakeholders often have cross cutting concerns and implications. We do continue to document these various aspects of both enablers and challenges, some of which are included in Section 2, and other are documented in detailed meeting minutes.

Therefore, the overarching research question is:

Is it **technically feasible** to use model-centric engineering in order to achieve at least a **25 percent reduction in the time** it takes to deliver a large-scale air vehicle weapon system, and secondarily Can we **radically transform** the way that NAVAIR and all contributing stakeholders operate in conceptualizing, architecting, designing, developing, producing, and sustaining systems and SoS

It is acknowledged that there are many possible hurdles beyond technical feasibility (e.g., organizational adoption, training, usability, etc.), but they have in general been reduced in priority for this phase of the effort. However, as our findings suggest that model-centric engineering use is accelerating, this apparent fact has surfaced another question (paraphrased from sponsor):

If we are going to rely more heavily on model-centric engineering, with an increasing use of modeling and simulations, how do we know that models/simulations used to assess "performance" have the needed "integrity" to ensure that the performance predictions are accurate?

There are four key tasks, which are described in this report, but a key decision made during the kickoff meeting was to attempt to "model the Vision" the rationale being that:

> If it is technically feasible "do everything with models" then we should be able to "model the Vision."

Surprisingly, we have heard people discuss the notion of a "Vision," but not much in terms of a modeling such a vision, with one possible exception, which we will discuss in Sections 2 and 5. In attempting to produce such a model, we have had some challenges in selecting an appropriate modeling approach that can be understood by most stakeholders, where a significant number have not done much modeling. Therefore, our sponsor has recently refined that objective, asking us:

> What is the "end state"?

This would reflect on the "Vision" concept, but also the operational aspects of government organizations like NAVAIR interacting in a more continuous type of collaboration with industry stakeholders; this is where the notion of the radical transformation comes into play. While

model-centricity may enable improved automation and greater efficiency, NAVAIR seeks its use in an improved operational paradigm.

There are many additional research questions that we discuss in this report, such as:

What are the emerging technologies and capabilities that will enable this Vision?

How will such a Vision work in the face of complex, human-in-the-loop, autonomous, and adapting system?

Can such approaches work in the face of safety and airworthiness requirements?

What are the technical gaps limiting the Vision?

What are the approaches to deal with risk when making decisions based on models and the associated simulations, surrogates and analyses?

Finally, there are some things that will likely be challenging to model, at least for now (e.g., human cognitive system interactions), and therefore there will be a risk framework integrated with the Vision. This risk framework will leverage other types of predictive models (e.g., stochastic) and methods to both embed decision-making knowledge and formalize both quantitative and qualitative information to support risk-informed decision-making. We have evidence of at least one approach that forms the basis for the risk framework, which we believe can begin to address the question about assessing the "integrity" of model predictions. We will discuss this in Section 6.

SCOPE

Given this context, we have been directed to reduce the scope for this effort to focus on the front-end of the lifecycle from pre-milestone A to critical design review (CDR). This is typically considered the front half of the "V" model. However, as is discussed in this report, many of our meeting discussions go well beyond this scope, as we consider the potential impacts that models or digital assets will have on the other phases of the lifecycle. We do document most of these potential ideas as they can possibly play a role in a radical transformation.

RESEARCH SUMMARY

This section provides a summary of the findings, results and next steps of this research as the final deliverable for phase 1 of RT-118. Sections 3 through 6 align with the task ordering shown in Figure 1, and provide additional information. This section presents the following information:

- What we mean by model-centric engineering and the model lexicon status
- A discussion about the Vision concept and enabling technologies
- Summary of and perspectives on the visits to industry, government, and academia to seek out the most advanced holistic uses of model-centric engineering
- Perspectives on what we have heard from the visits as it relates to the "technical feasibility" of using model-centric engineering
- Discussion about some challenges areas
- Scenarios for addressing the 25 percent reduction of time for development of a 5th generation air vehicle system that continues to increase in complexity
- Summary and next steps

MODEL LEXICON STATUS

Modeling terminology can be confusing, and we created a model lexicon (Task 2). However, a simple definition is not always adequate as there are many overlapping definitions. Some of the terms are overloaded. While we did give some references and example uses in our lexicon, they do not necessarily completely convey the broad concepts such as model-centric engineering. Other organizations (e.g., NASA) are working on similar definitions and groupings of terms, and we agreed to work more collaboratively on the lexicon.

Of particular note is that this task was characterized under the term Model Based Systems Engineering (MBSE), and we have repeatedly stated that NAVAIR means MBSE in the broadest way; as such, we and NAVAIR have adopted the term model-centric engineering. NASA was one of our first discussion meetings and they used the term model-centric [5], and other have adopted the term model-centric engineering too [76].

Status Task 2: we have captured over 300 named lexicon items related to the term "model," including levels, types, uses, representations, standards, etc. The details are described in Section 4; we have delivered these model-lexicon artifacts to NAVAIR for them to post internally.

VISION MODEL CONCEPT

At our RT-48 kickoff meeting, the question was asked "is it technically feasible to model everything?" As a result, we said that we would attempt to model the "Vision." This effort is part of Task 3, and we have identified a few similar examples that are discussed in working sessions. Two things have resulted from our efforts in researching what a "Vision" model might be, and how it might be represented:

We have seen only a few example fragments of Vision-like models [5]. Organizations typically model only the systems they want to develop and evolve. Often organizations do not think about modeling all of the elements of the environment, referred to by Ackoff as the **containing system** [1], the elements and interactions of the **designing system** [1], including elements of existing system, subsystem and parts, in order to create an instance of the target system, which would then be stored in a version of what some call a "System Model" [91]. (See Section 5.2 for a more detailed perspective on the containing system and designing system).

One organization has created at least a start of something that relates to the Vision [5]; they are using the System Modeling Language (SysML) [70]. We started an example SysML model to illustrate this concept too. However, in some of our working sessions, we found that not everyone was familiar or comfortable about these perspectives using SysML modeling **views.**

Therefore, we refrain from using those types of views in this section of the report. Instead, we will provide some examples to help with clarifying the concept of what we think should be included in the Vision.

WHAT IS A MODEL?

We have also heard from our stakeholders that some people may not understand what is meant by the term model, as well as having a consistent view on model-centric engineering. We are going to provide some details before moving on to the concepts involved in the Vision.

Modeling, in the broadest sense, is the cost-effective use of something in place of something else for some cognitive purpose. It allows us to use something that is simpler, safer or cheaper than reality instead of reality for some purpose. A model represents reality for the given purpose; the model is an abstraction of reality in the sense that it cannot represent all aspects of reality. This allows us to deal with the world in a simplified manner, avoiding the complexity, danger and irreversibility of reality [79].

George E.P. Box, said: "Essentially, all models are wrong, some models are useful." [21]

One key aspect of models and modeling is **abstraction**, which supports communication through different views with various levels of details. Details of importance can be emphasized while other details are not described. Most of us have been exposed to models for a long time, for example, a mobile of the solar system, as shown in Figure 2 shows the number of planets and might show the relative position of the planets, but it does not accurately show the plant's size or distance from the sun. Different views can provide alternative and relevant perspectives on the planets of the solar system and emphasize the relative size of the planets. To get an accurate perspective of a problem or solution often requires several **views** with some type of formal description of the relationship between the views. For example, the distance from the sun to each planet needs to be described using consistent units (e.g., miles).

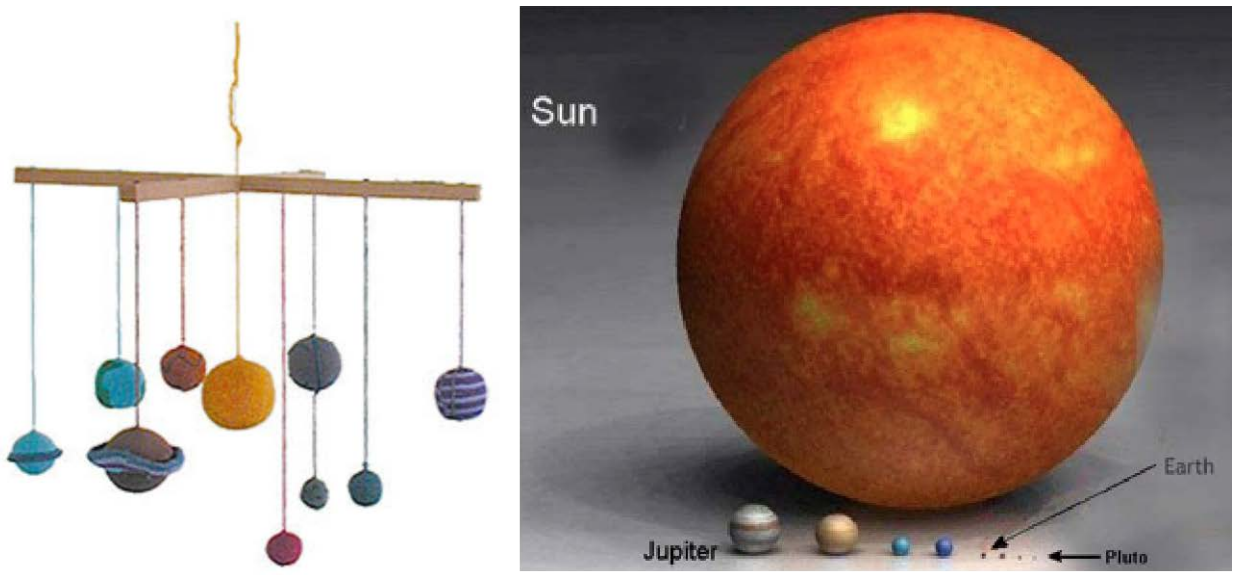
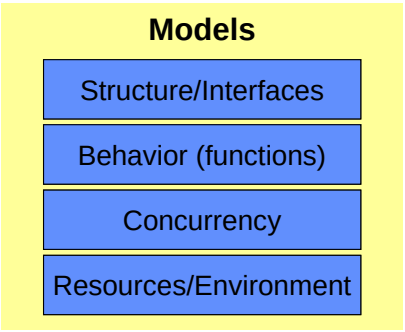


Figure 2. Two Model Views: Mobile and Relative Size of Planets[1]



Model-centric capabilities to achieve NAVAIR's vision will heavily rely on computationally-based models. We need to use the **relevant abstractions** that help people focus on key details of a complex problem or solution combined with **automation** to support the simulation and dynamic analysis of both the problem and solution, along with the mechanism for combining the information collected from the various abstractions to construct a system correctly. Some of the key abstractions can be categorized into types, such as:

Structure – 1D, 2D, 3D models, systems, subsystems, components, modules, classes, and interfaces (inputs and outputs)

Behavior (functionality)

Timing (concurrency, interaction)

Resources (environment)

Metamodels (models about models)

Many of these model-centric abstraction concepts have existed and evolved from programming languages, but within a programming language the combination of these views may be lumped or tangled together (e.g., spaghetti code). Most dynamic model capabilities cannot effectively leverage simulation, analysis or generation capabilities if the models are constructed in an ad hoc way.

Modeling methodologies (beyond process steps) are needed to guide the structuring of models to provide a means of systematically separating these views, because certain types of models are

[1] Image credit: www.thisisauto.com/.../wa07005i/l/ID_mobilel.jpg, http://elronsviewfromtheedge.wordpress.com/2006/08/23/and-you-thought-size-mattered/

constrained to permit only certain types of information. Model-centric automation relies on automated means for analyzing the views, deriving information from one-or-more views, and ultimately pulling sets of views together correctly to produce some type of computationally-based system, simulation or associated analysis artifacts and evidence.

OPERATIONAL PERSPECTIVE OF MODEL-CENTRIC INTEGRATION

Model-centric views provide a means to integrate different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity. Figure 3 provides an example documented in a case study that was published in 2008 [44]. While this was possible then, it does not go nearly as far as the Vision we believe NAVAIR is seeking. Hidden behind the scenes, there was manually created code to integrate the levels and views. This reflects on the types of software skills that will be required to assemble model-centric simulations for analysis until we improve the integration and interoperability of models (see Section 2.5 for discussion on challenges).

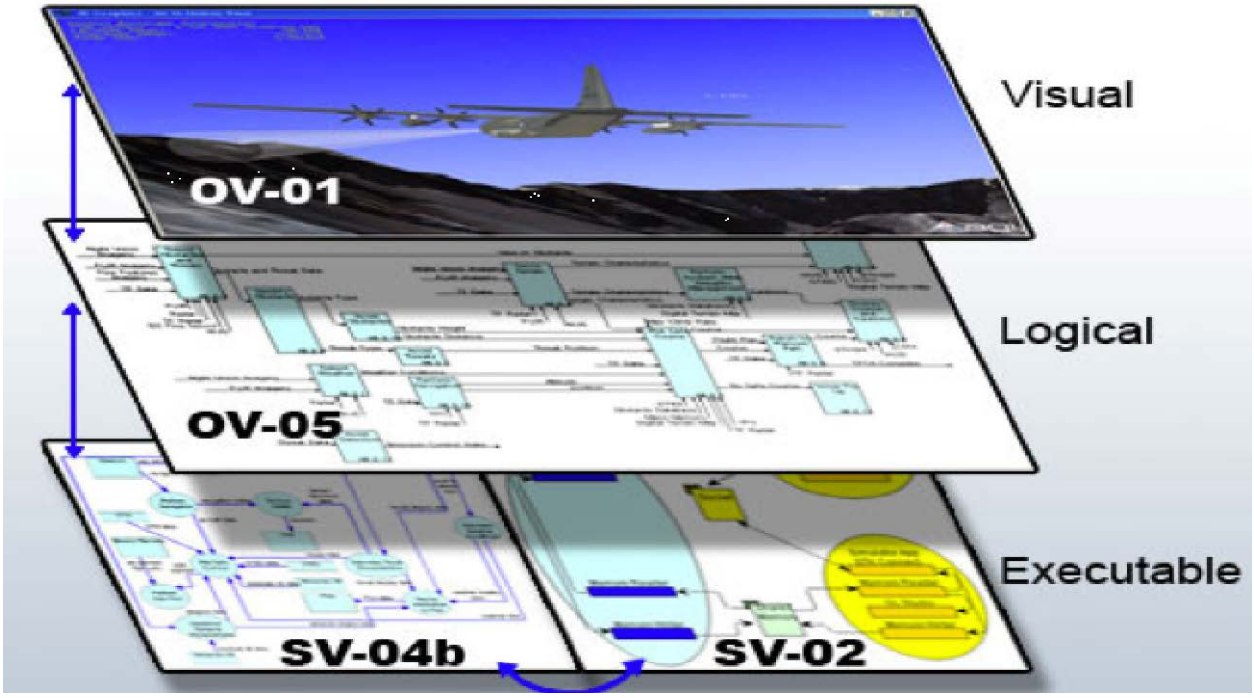


Figure 3. Model Centric Provides Digital Integration Between Views

Extending the previous example and relating it to a scenario of moving through the lifecycle phases, our team provided another representation that was included in the RT-48 final technical report [15] that extends this concept and relates to the lifecycle phases. This example is also abstract, but reflects on a NAVAIR objective, which is to continuously "cross the virtual V" early in the lifecycle in order to better ensure that the system design meets the SoS mission needs, essentially collapsing the timeline. Consider the following scenario using the typical release phase reviews as the time points to represent a notional timeline moving from left to right (e.g., System Requirements Review (SRR), System Functional Review (SFR), Preliminary Design Review (PDR), Critical Design Review (CDR)).

In a model-centric engineering world the models at SRR would reflect on the new high-level aircraft needs/capabilities, as conceptually rendered in Figure 4; there is likely a strong relationship between these new operational capabilities and the mission needs. These models would need to be sufficiently "rich" that we could computationally connect them to other surrogates, such as software components (new/legacy), hardware and physical surrogates (e.g., previous generation aircraft). We ideally want to have some type of dynamic operational capability operating from the very beginning of the effort (all digitally). As we transition through the lifecycle phases SRR, SFR, CDR, and PDR, we would use a similar process on lower-level models and components that provide increasing levels of fidelity that is moving us from the analysis of the problem and aircraft needs and closer to the actual system as the decisions for the physical design are defined and refined. We begin to focus on detailed functional and timing behavior, with models that predict performance characteristics and begin to clarify the margins and uncertainty; we would continue the transition from the use of surrogates to versions of the implemented design. As we continue to move through the acquisition phases to CDR, especially for 5th generation air vehicle systems, we will have much more software than ever before, including software that connects models with the simulations, surrogates, components and live or historical environmental data.

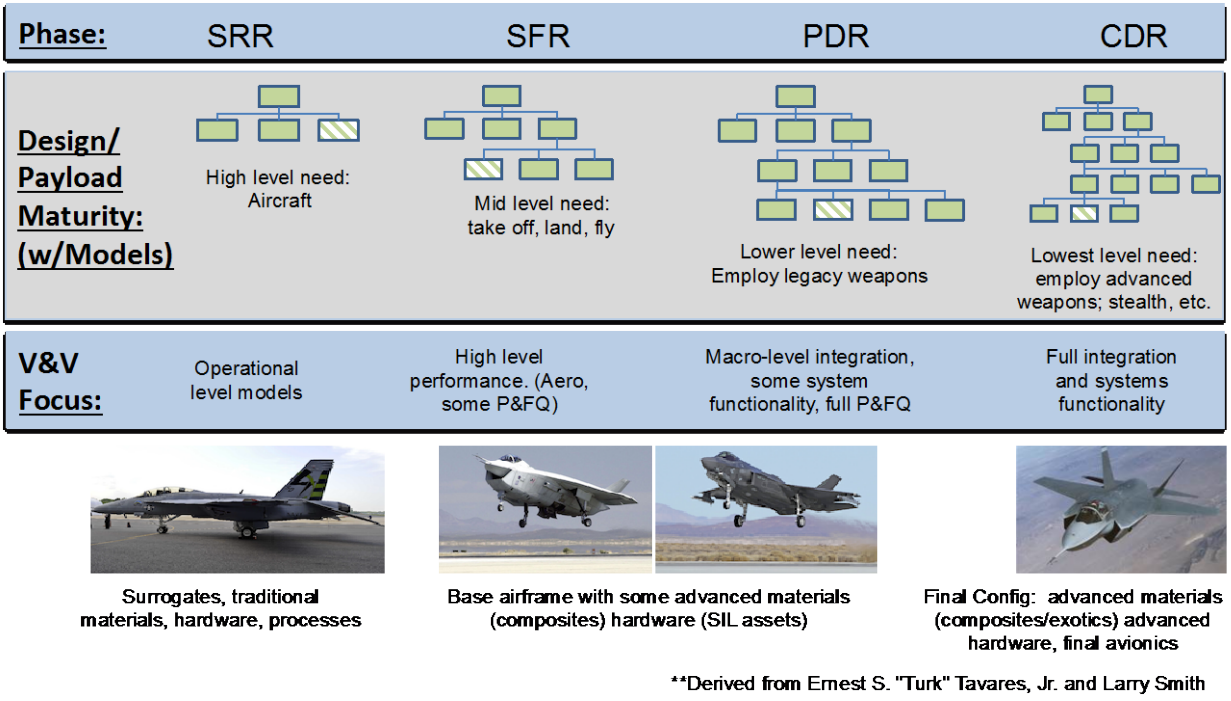


Figure 4. Dynamic Models and Surrogates to Support Continuous "Virtual V&V" Early in the Lifecycle

Increasingly there will be much more complexity in the software prior to PDR and CDR, and this creates different concerns for NAVAIR from prior generations of air vehicle systems. Testing will be required to ensure that the continuously refined representation of the system models and implementation meet the timing (temporal interactions) and performance needs.

Challenge: Given this scenario, what is NAVAIR's role? Does this perspective have any negative impacts on the "technical feasibility," or is this just some type of operational

perspective. Does NAVAIR build the models? Do they work with the contractors who build the models and integrate to surrogates and a refining solution? We raise these points, because it does have an implication on the "Vision" model and "End State," which is additionally clarified in Section 2.3.

VISION, "AS IS" MODEL AND AIRWORTHINESS PROCESS

Task 3 investigates how to model the "Vision," "As Is" and the Airworthiness process. This is a joint effort with:

"As Is" process model being worked by Ron Carlson and Paul Montgomery, from Naval Postgraduate School (NPS)

Airworthiness aspects being worked by Richard Yates and Rick Price from MITRE, and Brian Nolan from SOLUTE

Vision being led by Mark Blackburn with contributions by our collaborators and many others

We typically hear about a System Model [5], [91], which should ideally represent all the data and information to produce the target system, as well as all of the evidence that characterizes the consistency, completeness, and correctness of the information. In this case, it is scoped to the Program of Record (POR) for our task. The information to cover what we believe to be the Vision should include:

Sufficient information about the **containing system** (information about the problem and environment) [1]

- This information should come from the mission analysis as sets of desired operational capabilities with performance objectives
- NAVAIR is conducting some similar type of research effort at the mission-level
- This is briefly discussed in Section 5, but a high-level perspective is provided in Section 2.3.1

All the information about the **designing system**

- Every tool for model creation, storage, simulation, analysis and their interconnections that is used to create, simulation, or produce analysis information related to decisions or dependent information
- One organization has an organization that develops the enterprise, which is the system for producing the target system
- We give an example in Section 5

All other platform related information that provides some aspects of the interrelated capabilities associated with the POR (**system instance to be designed**/evolved), including revisions, variants, and even trade spaces analysis results from design alternatives not selected

Some of these perspectives are provided in Figure 5[2]. This figure puts into perspective the mission capability threads that have relationships to different PORs for the different platforms, and puts

[2] These figures come from a briefing given by Jaime Guerrero that is approved for public release; distribution is unlimited.

into context some of the aspects of the containing system (i.e., the interrelationship to other PORs in order to support a capability). This image abstractly reflects on the information about the existing assets of previous systems (platforms) that can play a role in model-centric engineering:

These would be represented in some type of reference model or master template[3]

- All of the existing elements (components) that could be put into a system derived from historical knowledge and reuse
- The relationships (valid connections)
- Model representation of new elements (components) from new design ideas and/or technological advances

These provide the building blocks for defining and refining a new capability

Currently, this information is largely defined in documents; it may be partially modeled, and/or held by contractors (data, models, knowledge)

Therefore, in order to realize the Vision, NAVAIR going forward needs total and continuous access to this type of information, but in a better form than documents, such as models.

[3] A commonly used term is reference architecture. A reference architecture in the field of software architecture or enterprise architecture provides a template solution for an architecture for a particular domain.
Instance: An instance is a specific system that can be developed using the template of the reference architecture. Any new technology advances should be incorporated back into the reference architecture for reuse in the future.
Variance: Items that meet the same definition in the reference architecture but have different solutions.

Figure 5. Putting the Vision into Context

CONTAINING SYSTEM

The Containing System must represent the SoS, including environment and resources with sufficient fidelity and semantic precision to understand how a target system interacts within its environment. These types of view are needed to understand the problem and to investigate, through models, the different alternative systems that can address the problem. In general, a complete high-fidelity representation is not possible, therefore there will be some type of abstraction of the containing system such as reflected in Figure 6. This is one scenario of a capability, and the particulars of the interface parts can include the environment, such as the ship, landing surface, arresting system, weapons, weather, threat types, operators, etc.

Figure 6. Perspectives on Aspects of the Containing System[4]

Some of the current approaches to modeling used by NAVAIR such as DoDAF models, are static and do not capture enough information to support simulations. They are semantically imprecise when it comes to representing behavior and the temporal (timing) interaction to be able to assess and predict the needed performance. Our field visits to commercial and industry organizations, discussed in Section 2.4, reflect on modeling and simulations capabilities that are neither well integrated nor interoperable, but some organizations are integrating mission simulations with system products, components or other simulations. While the interest and intent exists, the standards do not keep pace with the technologies. We heard that there is an inadequate level of cooperation to foster the needed integration and interoperability. We make this point, because it is important to the overarching research question.

Challenge: There is currently a lack of cross-domain model interoperability, consistency, and limitations transforming models with the required semantic precision to provide accurate information for decision-making.

Implication: Without cross-domain integration and interoperability it is difficult to assess the cross-domain impacts, and makes it difficult to understand the uncertainties, which is related to our sponsor's question about model "integrity."

[4] bigstory.ap.org

DESIGNING SYSTEM

The Designing System is the entire enterprise, which includes model capture, synthesis, simulation, analyses, testing, configuration control, workflow automation, product lifecycle management (PLM), etc. The idealized goal is to transform all information so that it can be used for simulation, synthesis, analysis and ideally "models for manufacturing," (e.g., "3D print the system"), models for training, operations and sustainment. This is not realistic for the entire system or all of the parts, at least not today, but our sponsor desires it as part of the Vision and End State.

We have had a number of discussions with industry suppliers and users of technologies, and there are technologies that support multi-domain and multi-physics-based 1D, 2D, and 3D modeling, analysis and simulations, but we need go beyond.

Implication: NAVAIR has options, as there are competing commercial suppliers; using different tools provides a degree of independence that can be used to substantiate model "integrity" arguments

Challenge:

- The emerging capabilities often outpace the standards, and impact integration and interoperability between different vendor capabilities.
- Does this make it harder for NAVAIR, because they cannot impose one set of tools; do they need to maintain all?

INDUSTRY, GOVERNMENT, AND ACADEMIA VISITS

This section summarizes some information about the visits with industry, government, and academia. Prior to each meeting, we send our coordinator package to an organization coordinator; the package explains the overarching goals of the research task. We often iterate with the organization about an agenda. In coordinating the agenda with our organizing hosts and at the start of each meeting we usually posed the question:

> "Tell us about the most advanced holistic uses of model-centric engineering you have seen in use on projects/programs or that you know about"

We also state the question posed by our NAVAIR sponsors:

> Do we think it is "technically feasible" for an organization like NAVAIR to have a radical transformation through model-centric engineering (everything digital) and reduce the time by 25 percent for developing and delivering a major 5th generation air vehicle system?

Table 1 provides a status of the discussions as of 31-December, 2014; this does not include several follow-up discussions. The discussion meetings summarized in the third column typically have occurred at organizations' business operations or at NAVAIR. Discussions are often one-day meetings with presentations, heavily discussion-based with some demonstrations.

Table 1. Discussion Group Summary

Discussion Groups				
Category (needs discussion)	Preliminary Held	Discussion Meetings Held	Coordinated	To Be Coordinated
Commercial – provides tools and/or service to produce systems	2	4		
Industry - produce systems	3	12		
Government	2	5		1
Academia		1	1	

Most of the discussions with industry and commercial organizations were governed by some type of Proprietary Information Agreement (PIA) or Non-disclosure Agreement (NDA). In addition, some of the provided material is marked in a manner that limits our distribution of the material. Due to the need to sign a PIA/NDA, we are being careful about disclosing those organizations in this report. In addition, because we cannot disclose information about commercial or industry organizations, we are limiting how we discuss the other organizations too, and will use and reference only published and publicly available information.

We have created meeting minutes, which generalize the information we heard during the discussions. NAVAIR wants to share it with our NAVAIR research team, therefore we are including

the following notice on meeting minutes that are distributed to our team, per Jaime Guerrero, Director, SEDIC - AIR-4.1, NAVAIR:

> DISTRIBUTION STATEMENT D. Distribution authorized to the Department of Defense and U.S. DoD contractors only. Other requests shall be referred to SEDIC Director, AIR-4.1, Naval Air Systems Command, 22347 Cedar Point Rd., Bldg. 2185/Rm. 3A54, Patuxent River, MD 20670 - (301) 342-0075.

These meeting minutes are not part of the official RT-118 deliverable, but because this report is an official deliverable and will be publically available, we are not going to include any information about the organizations that we met with, rather we provide a generalization through the following narrative and discuss the results in the aggregate.

DISCUSSION PERSPECTIVES (ANONYMOUS)

This section provides a summary of the wide range of information we discussed in the meetings. Some of the meetings were scoped to particular disciplines and domains that would support the overall NAVAIR objectives, as there are often many areas of expertise required to cover the engineering efforts of an entire system. We are going to present the summaries in a top-down manner starting from a mission-level scenario.

MISSION-LEVEL SIMULATION INTEGRATION WITH SYSTEM SIMULATION AND DIGITAL ASSETS

Several organizations discussed mission-level simulation capabilities, but one organization demonstrated mission-level simulations that are being integrated with system simulation, digital assets and aircraft products providing new type of web-based services:

> We attended a live (with some artificial data) multi-scenario SoS demonstration that runs on a modeling and simulation (M&S) infrastructure that integrates with other M&S capabilities as well as live products that can be hosted within a cockpit or operate through server and web-based services
>
> The scenarios represented commercial version for DoD-equivalent mission analyses
>
> The M&S infrastructure is used to both analyze new types of services that can be added to their portfolio, but is integrated with other existing systems and can be populated with real or artificial data
>
> These capabilities are used in a way that improves their own systems and capabilities, but they use these capabilities to solicit inputs from potential customers on new types of products and services
>
> - A scenario was provided about capabilities that are part of a services platform to support logistical planning through real-time operations and maintenance
> - New digital products such as real-time health management that integrates through web-services connecting pilots in the air with maintenance operations at the next airport

However, even with the advancements this organization discussed some challenges with developing the integrations as there was not a grand architectural scheme or plan when they first started developing the underlying infrastructure

- We have heard this type of story several times from other organizations too
- This is the challenge for both representing simulations of the Containing System, and then integrating them through the Designing System, existing systems and components, and new model-centric designs

Implication:

- Companies are advancing the types of technologies needed by NAVAIR, because they are leveraging new business opportunities out of some of the enabling technologies of yesterday
- The integration to make it a reality is still challenging, as they are trying to leverage existing (legacy) systems that were not necessarily designed to be integrated
- This example not only addressed part of the scope of our research task, but truly demonstrated several aspects of the concepts of model-centricity
- This example provides some type of relevant information for the other NAVAIR initiative scoped at the mission-level

---

3D ENVIRONMENTS AND VISUALIZATION

Several organizations demonstrated (or showed results from) some of their 3D and visualization capabilities:

One organization discussed and demonstrated the use of two different types of 3D environments for customer engagements, but also for on-going (often daily) design engineering analysis and review sessions in 3D environments

They do use commercial technologies, but have developed a significant amount of infrastructure on their own

We heard similar stories from others about the need to develop their own infrastructure

We also visited the Cave Automated Virtual Environment (CAVE), as shown in Figure 7, where we were immersed in a virtual 3D environment that is used for both analysis and design review purposes [4]

Implication:

- These capabilities provide a collaborative environment not only for design, but for continuous reviews
- This scenario aligns with a concept we continually discuss, and provides the type of environment that could enable a “radical transformation” in the way that NAVAIR operates with its stakeholders
- If NAVAIR is going to integrate and/or transform the System Engineering Technical Review (SETR) process (see Section 5.3.1.4) to leverage these types of capabilities, they may need to define methodological guidance to align with a model-centric

approach to be able to continuously capture the evidence and actions produced from these types of environments and engagements in continuous and collaborative reviews

Figure 7. Cave Automated Virtual Environment (CAVE)[5]

DYNAMIC OPERATIONAL VIEWS FOR MISSION AND SYSTEM SIMULATION AND ANALYSIS

There are modeling environments to create dynamic Operational Views (e.g., an OV1) to understand and characterize the Mission Context for the needed System Capabilities, as shown in Figure 8. In traditional DoDAF models, we are used to static Operational Views (OV1), but the newer capabilities provide for dynamic operational scenarios not only allow for analysis, but they are being leveraged as scenarios for testing. In many instances these types of capabilities have integrations with other types of models, simulation and analysis capabilities, similar to Figure 3. This example comes from an organization[6], and while we did not speak with them in any of our discussion meetings, we have had a number of interactions with them through Stevens Institute of Technology on other research tasks. Many of the organizations as well as NAVAIR use the tool kit, which has an evolving set of libraries that can be used through their platform to support dynamic visual-based analysis. The example discussed in Figure 3 used this tool kit at the OV1 level.

Implication:

- Model-centric engineering is moving beyond static DoDAF views

[5] Image credit: image credit: media.gm.com

[6] Being consistent with our goal to not single out any organization, we provide an image credit, but will not mention this company directly. Note: this is not the only product in this space.

- The computational and visualization technologies bring the behavioral views into perspective, but can increasingly bring the temporal aspects into play
- This provides much more information to support decision making
- This is a capability that can be used at the mission level as well as the system level

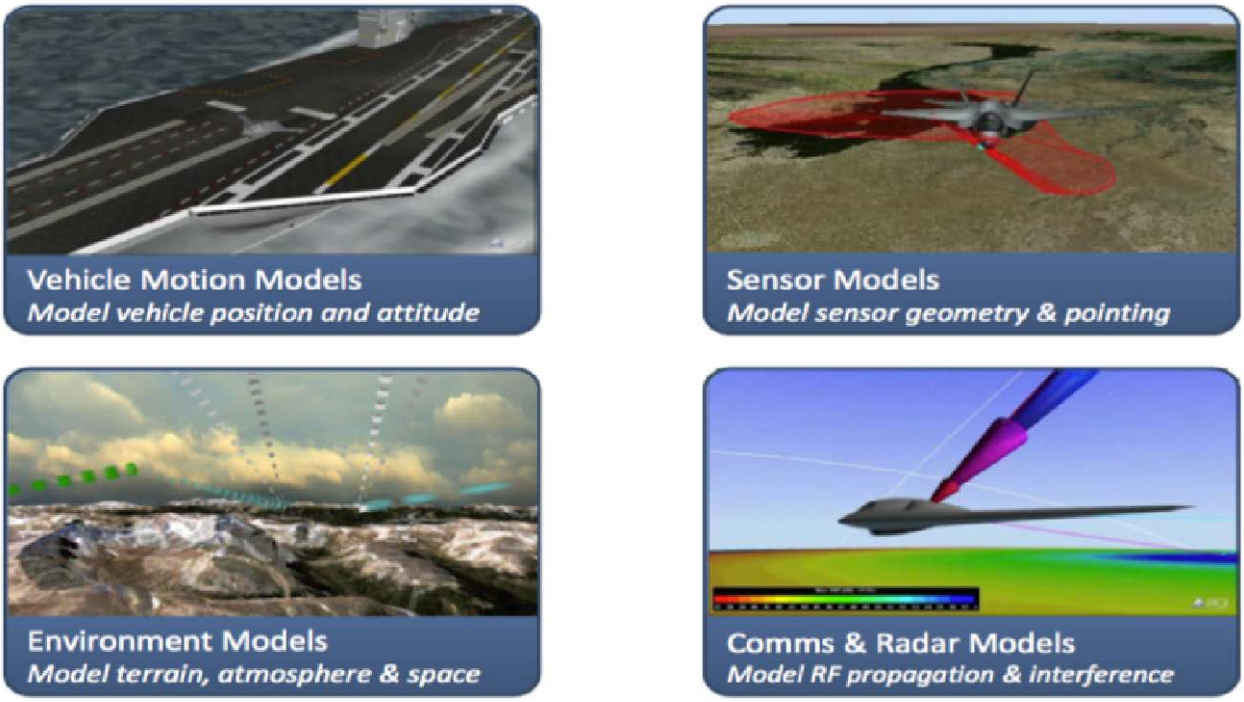


Figure 8. Dynamic OV1 with Integrations to Other Models and Digital Assets[7]

MULTI-DISCIPLINARY ANALYSIS AND DESIGN OPTIMIZATION (AKA TRADESPACE ANALYSIS)

We had a presentation and demonstration in a Multi-Disciplinary Analysis and Optimization Laboratory. This organization mentioned that several years back they had a consulting organization assess their state of the practices against the other contractors, and it was believed that they were trailing others in this capability area. They decided that they did not need to do a Return on Investment analysis, and just moved forward with putting their lab together. The information they presented showed that they have a much more comprehensive approach today, not only the integration of the tools, but the methodological approaches. Here are a few points:

They established the information technology (IT) infrastructure to facilitate the integrations across the design space

- Integration of many facets Aero, Mass Properties, Performance, Propulsion, Operational Analysis, Ops-support, Manufacturing, and assembly and lifecycle costs across multi-mission scenarios, but not necessarily cross-domain at the same time

[7] Image credit: AGI

The power of the tools often allows them to spend more time doing more in-depth analysis

- They are systematic about creation of design of experiments
- With the more powerful tools, the engineers often perform more excursions of the tradespace
- It is anticipated that this could lead to more robust designs, with fewer defects helping to reduce cycle time
- They stated that in the use of these technologies, they often uncover or expose things that are not intuitive – that is the more comprehensive analysis allows for many more excursions and they can uncover issues early
- While we ask organizations for measures, organizations are not that willing to provide metrics, other than anecdotally

The approach reflected on methodological rigor

- We have pointed this out many times that through the use of these type of model, simulation and analyses that methodological guidelines need to be more strongly defined than simple processes in order to leverage the tool (or toolchain) capabilities
- Their approach frames the system both top-down and bottoms-up, allowing the mission analysis to help in supporting the physical-based analysis/tradeoffs
- They use off-the-shelf tools with their own customizations, like many others

The lab facilitates collaboration

- The lab environment tends to draw people into use the capabilities
- "Junior folks" gravitate to this environment
- The junior engineers have the computational and software experience needed to use and extend these types of capabilities

Implication for NAVAIR:

- In the aggregate, we saw the use of design optimization for tradespace analysis that is increasingly cross-domain, including aspects such as cost models
- We may be at a tipping point where we can have the appropriate types of collaboration with industry providers through the use of models and simulation; industry seems to be in favor of this
- We may be able to have greater trust in the models, and do the continuous reviews directly with the models (as stated before: a single source of truth)
- The pragmatics may boil down to contracts, policy, culture, and intellectual property

---

1D, 2D, & 3D Model Creation, Simulation, Analysis for Physics-based Design

We have heard from many organizations that discuss 1D, 2D and 3D model creation, simulation, analysis, and management capabilities focused primarily on physics-based design, with increasing support for cross-domain analysis:

Most are focused on physics-based models

Some have unique capabilities and there is an increasing trend/push to support broader cross-domain analysis through better integration of different types of models

Some allow for the plug-in of niche-capability libraries and solvers, using a platform-based approach to create more of an ecosystem (i.e., "think apps")

Some are customizable to leverage High Performance Computing (HPC)

- That is, they have been programmed to take advantage of parallel computation
- While this is typically assumed, it is not the case – we spoke with organizations that stated that organizations may use HPC, but the simulations/analyses are not always programmed in a way to take advantage of the speed derived from parallelism

There are challenges in model transformation and/or interoperability, and the need for formalized semantics is known

There are also multiple suppliers that often provide a suite of tools that cover some portion of the lifecycle

- Some of the commercial organization acquire companies/products to better complete their portfolio of tools; however, we know that just because a company acquires a product that there is not necessarily a seamless integration with the other products

Implication:

- These physics-based capabilities are necessary, but not sufficient as many of these capabilities still do not deal with software; multiple organizations acknowledged the significant challenge of software that continues to increase in complexity; see Section 2.5.1 for more details
- Without better model integration, interoperability, and/or model transformations, how is NAVAIR going to deal with so many types and variants of models
- With so many modeling and simulation capabilities, we return to the question of "how do we know the integrity of the model predictions," we will discuss this in Section 6

MODELING AND SIMULATION INTEGRATION WITH EMBEDDED SYSTEM CODE

There were many relevant topics that support the vision of model-centric engineering, including one discussion by an organization that performs modeling and simulation of the flying qualities that integrate directly with the code generated from the Simulink model for the control laws of an actual aircraft.

PLATFORM-BASED APPROACHES

Platform-based approaches are used not only by the commercial tool vendors as discussed above, the developer of systems have been improving their approach to use platform-based approach and use virtual integration to help refresh systems and do system upgrades on periodic schedules, which in many cases is business critical:

We heard from two organizations in the automotive space discuss platform-based approaches that are tactically driven by the periodic cycles demanded for sales roll outs at 12, 18, 24, 30, and 36 month delivery cycles

- 12, 18 months - they might change feature colors, but every interface is exactly the same, and no electric changes
- 24 may make some minor changes, and electrical
- 30 change types of subsystem, component (e.g., Figure 9, based on approach that uses the Modelica [70] and Functional Mockup Interface (FMI) standard)
- 36 months – complete redesign

For longer cycles times, they use a "W" lifecycle metaphor rather than a "V," where the first part of the "W" might cover more of the prototyping efforts, and the second part of the "W" is more related to verification of the implementation

Unlike NAVAIR, they completely decouple research and development (R&D) from development (even the "W"). This means that some of the key aspects of what we are looking at for this research project, from pre-milestone A to Critical Design Review are not part of their typical roll out process

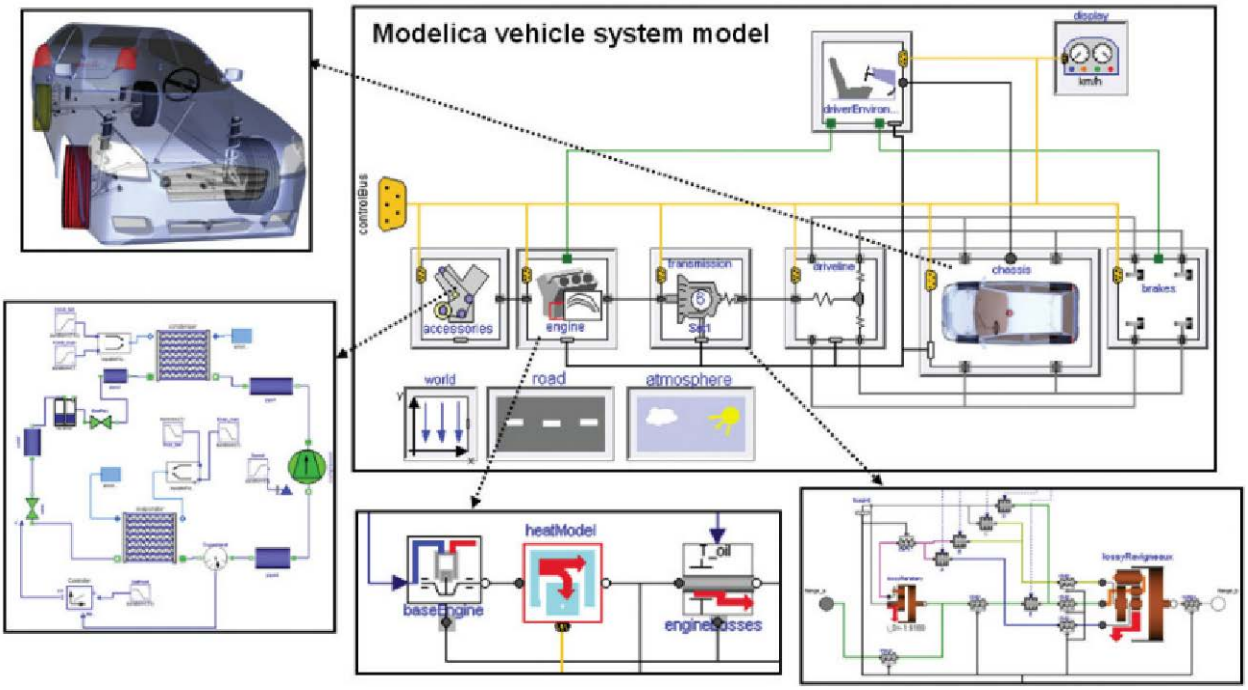


Figure 9. Vehicle System Platform-based Model

MODELING AND SIMULATION REDUCING PHYSICAL CRASH TESTING

The automotive companies stated that modeling and simulation is being used to significantly reduce crash testing. Some mentioned numbers such as from 320 crash tests to 80 crash tests. This is of particular interest to NAVAIR as they want to assess designs earlier and more continuously by flying virtually, as flight-testing is expensive. In addition, it can be extremely challenging to obtain a flight clearance unless all of the airworthiness criteria are provided.

WORKFLOW AUTOMATION TO SUBSUME PROCESS

We have continued to have conversations about workflow automation. Automated workflows arose from the manufacturing world, and the key idea is that if we could formalize all the model-centric artifacts, including the process, we could "generate" information to drive a workflow engine that would completely subsume the process. This would get NAVAIR to the place where decisions were completely data-driven, and the process would be autonomous and adaptive, and coordination would replace planning.

Workflow automation has the potential to subsume the entire process; everything driven by data, data dependencies; this would be towards a "radical transformation"

- The key reason for this concern/question is that the effort in modeling the "As Is" process is reflecting that it is potentially too difficult to ever fully create or follow a document-driven process

There are workflow engines to drive the enterprise operations, but when we asked the question "do you use a model to generate the information to drive your workflow engine," they said:

- "No, but that is a good idea"
- It seems that most workflow engines are programmed manually
- One organization has started to develop capabilities of modeling the workflow

To a lesser degree, there are other types of products that provide workflow automation support integrations for work such as design optimization (see Section 2.4.1.3)

- We spoke with both commercial companies that provide these capabilities and industry companies that use these technologies
- They do help speed up and make the design optimizations more efficient, and allow for more iterations, and more systematic regression analysis

Implication:

- In the future, we think that a Vision model would allow workflow engines to completely subsume the process and drive every decision based on real-time data; this could completely subsume the current SETR and fit in with the new SETR Manager, which is discussed Section 5.3.1.4
- This will be critical as the current process cannot adjust quickly enough to adopt new technologies that will be needed to keep pace with NAVAIR's need for speed to address continuously emerging threats

PRODUCT LIFECYCLE MANAGEMENT

Holistic approaches invariably bring in the need for some type of product lifecycle management (PLM) so that every piece of data/information is captured and traceable from design through manufacturing and possibly training. While this again might seem slightly out of scope for the

objectives of our research task, we briefly report on this, because it was covered in a number of discussions with other stakeholders from NAVAIR who are directly concerned with the need.

We have heard two of the large commercial companies discuss their myriad of products, including the Product Lifecycle Management (PLM) systems

Some companies said that the key reason for moving to PLM is for tracking every single item in a vehicle for warranties and recalls

If NAVAIR could characterize every type of data element required within a total "System Model," this would provide the schema for information that should and could be captured in a "PLM-like" system

This would/could be used to support every decision that needs to be made

- Every time an artifact was obtained a workflow engine could trigger automation of additional analysis or trigger individuals that a decision was needed based on new data

Implication:

- The Vision model concept is an enabler for the types of information that would need to be captured/stored by NAVAIR in order to make decisions
- Integrating the automated workflow and PLM-like systems, could put the information at the fingertips of the SMEs could work towards improving the speed of decision making
- Current PLM systems are not necessarily semantically rich-enough to support the vision concept

---

MODELING AND SIMULATION OF THE MANUFACTURING

Our sponsor also talked about models-to-manufacturing in our kickoff meeting, and while this may seem out of scope, the point is that organizations are simulating the manufacturing processes in advance of developing the tooling. In addition, the set-based design concept [82] originally attributed to Toyota described how the design and manufacturing processes work more concurrently. These concepts are strongly related to tradespace analysis and design optimization [44], as discussed in Section 2.4.1.3. This may also provide a means for reducing the time to develop large air vehicle systems.

One organization discussed model-based manufacturing, model-based inspection, design for manufacturability, additive manufacturing, their smart manufacturing efforts, and advanced design tooling (modeling and simulation infrastructure). They use an "In-Process Linked Models" methodology that provides an interesting metaphor that relates to the concept of a continuous PDR and CDR that we discussed at the kickoff meeting and many working sessions. They use modeling and simulation to analyze the manufacturing process before finalizing all of the details of the process and tooling; the very essences of models provide the ability to simulate the steps through design and manufacturing; this creates new ways to increase the assurance that the designed system is producible.

Implication:

These detailed types of consideration about tradespace analysis and modeling for manufacturing are often consider after CDR. How would NAVAIR operate considering this concept could have deep implications in speeding the delivery of capabilities to the war fighter.

DISCUSSION PERSPECTIVES (PUBLICALLY KNOWN)

This section discusses a few specific topics from organizations that we can discuss publically.

NASA/JPL – MODELING PATTERNS, ONTOLOGIES, PROFILES, CONTINUOUS INTEGRATION

NASA/JPL provided a perspective on their concept and evolving instantiation of a "Vision" model [5]. They have modeled and are formalizing the overarching Model-based Engineering Environment (MBEE) [35] (designing system) being used to develop instances of a system as well as the mission characterization that is captured in a system model. They continue to formalize the modeling methodology through model patterns [26] that are captured through ontologies, which are associated with a tool-based approach that not only guides development, but provides model analysis to ensure compliance with the patterns (e.g., models are well-formed, consistent, etc.) [52]. This provides their foundation for a single source of truth that is used both for development and continuous reviews.

Among other topics mentioned previously, NASA/JPL has developed an Architecture Framework Tool (AFT) for Architecture Description [6], which provides an overarching perspective on one of the views needed for our Task 3 vision model, and is partially supported with their evolving Open-MBEE [35].

These two concepts are further supported with a rigorous approach to systems engineering (SE); they have identified around 25 modeling patterns applicable to systems engineering. They formalize the patterns in ontologies using Web Ontology Language (OWL) [89] to provide a way of defining a set of concepts and properties applicable to the domain of discourse; in this case not about the space domain, but about the SE domains for concepts such as: component, function, requirement, and work package, data properties like mass and cost, and object properties (relationships) like performs, specifies, and supplies. This provides for a controlled vocabulary and enforcing rules for well-formedness, which permits, among other things, interdisciplinary information integration, and automated analysis and product generation. Because the SE ontologies are expressed in OWL, they are amenable to formal validation (syntactic and semantic) with formal reasoning tools. The approach embedded in SysML and the OWL ontologies are created by transformations from SysML models [52]. Once a model is completed other transformations are performed to the model, such as checking properties of well-formedness and consistency of the model. They currently have about 60,000 test cases for checking these types of properties. The approach is illustrated in several case studies [60].

Implication:

- These types of test cases fundamentally relate to one type of model measure that we discuss in Section 5.3.1.3

- While we were told that this is a work in progress this information provided the best story we heard as it relates to formalizing the concept of the Vision model (Task 3).

SANDIA NATIONAL LABORATORY

Sandia discussed some of the most advanced approaches for supporting uncertainty quantification (UQ) to enable risk-informed decision-making. The information they provide reflects on the advanced nature of their efforts and continuous evolution through modeling and simulations capabilities that operate on some of the most powerful high performance computing (HPC) resources in the world. We heard about their HPC capabilities, Common Engineering Environment, methodologies on Quantification of Margins under Uncertainty (QMU) [66] and an enabling framework called Dakota [80], which should contribute to our Risk Framework (Task 4). Sandia's team also discussed the various modeling and simulation capabilities and resources that run on the HPC. This ultimately led into a discussion about Model Validation and Simulation Qualification. More details are provided in Section 6.

Implication:

- This particular discussion provided some key information that could play a significant role in model validation
- Sandia uses these type of approaches, because they cannot perform tests such as nuclear device testing

DECISION FRAMEWORK

The Army's TARDEC provided a presentation and demonstration on an evolving a framework called the Integrated System Engineering Framework (ISEF) [41] [84]; this capability could provide a decision framework for NAVAIR. Briefly, ISEF is a Web-enabled collaborative decision-making framework to support knowledge creation and capture built on a decision-centric method, with data visualizations, that enables continuous data traceability. The framework integrates a number of different technologies that support decision-making applicable to different phases of the lifecycle, for example:

- Requirements – they have their own requirement management capability
- Feedback mechanism
- Portfolio management
- Architecture (through other MBSE tools)
- Tradespace analysis
- Risk
- Road mapping

While the information from this meeting may not directly address the research question for a radical transformation, the information we received seems valuable, as ISEF is complementary for a transitional model-centric approach.

ACADEMIC

We hosted a Panel Session at CESUN 2014, the 4th International Engineering Systems Symposium, June 11, 2014. Mark Blackburn was the moderator, and he aligned the theme of this panel discussion with our research task. We titled the talk controversially:

"Is it Technically Feasible to Do Everything with Models?"

We briefly summarize the event and information, because it was presented at a conference and is public domain. This conference is academic and we tried to get panelists that might provide a slightly more academic perspective, pointing to some challenging areas for future research. Briefly, we covered a few points:

Increasing complexity is a real challenge

One speaker (Stephan vanHorn from Honeywell) represented the Aerospace Vehicle Systems Institute (AVSI), which is a member-based collaboration between aerospace system development stakeholders that aims to advance the state of the art of technologies that enable virtual integration of complex systems

- The members include: Airbus, Boeing. U.S. DoD, Embraer, U.S. Federal Aviation Administration (FAA), Honeywell, U.S. NASA, Rockwell Collins, and Software Engineering Institute/CMU.
- The System Architecture Virtual Integration (SAVI) team believes a model-based virtual integration process is the single most effective step the industry can take to deal with complexity growth

Some high-level information about the responses from our panelists about question: "Is it Technically Feasible to Do Everything with Models?"

Axel Mauritz (Airbus Group) – No

- He confirmed the move towards the concept of a Reference Technology Platform for a platform-based designing system
- Final thoughts:
  - Not practical to do everything in models, for example: hard to represent "ilities"
  - Modeling efficiency - What is worth (value of) to model?
  - Legal question - How to sign a model?
  - Total system representation - How can we model, what we do not know (system interaction, unintended/ emergent behavior)?

Chris Paredis (National Science Foundation) – No

- He provided a good characterization for the need of precise formalism in models in order to address some concerns of semantics and model transformations
- Emphasized the importance of modeling based on the value (e.g., efficiency, reliability, performance, demand, cost) of the results
- Challenges:
  - Integration of different views
  - Integration of different formalisms

- Holistic elicitation and incorporation of knowledge
- Ontologies

Stephan vanHorn (Honeywell/SAVI) – "Never say Never"

- Described a Model Vision from the beginning through to the Supply Chain
- Work needed:
  - Integration of descriptive models (e.g., SysML) and executable models (e.g., Simulink)
  - Incremental certification using provably correct virtual design – address verification concerns for safety (e.g. Formal Methods and MBD annexes to DO-178C)
  - Sufficient system model fidelity to elicit emergent behavior to test for unknown unknowns

Some of the comments from our panelists provide a good lead in to the next section on some of the gaps and challenges associated with the "technical feasibility" question.

GAPS AND CHALLENGES

During our site visits, we asked the organizations to share some of the gaps and critical challenges too, and several of them we have been highlighting in the previous summary. Some provided inputs beyond the question of "technical feasibility," including some other good ideas, like the decision-framework discussed Section 2.4.2.3. We heard many different types of challenges such as:

Lack of **affordability** of projects and activities

Mission **complexity** is growing faster than our ability to manage it

Not identifying design or **integration** problems until late in lifecycle

- We emphasize integration, as the concept of cross-domain simulation from models has been pointed out before
- Complex systems have greater cross-domain dependencies, and many of the modeling and simulation efforts are not doing analysis in terms of the integration of models and their associated simulations
- In addition, we stress that once integration occurs, it requires more precise semantics: structurally, behaviorally, and temporally; these may be some of the biggest challenges related to the technical feasibility question for the research task

Having to **hunt** for data or supporting material during mission anomaly resolutions

Inability to share models in a collaborative environment

- This point again may relate to the underlying semantics of models in specific domains that are not easily shared

Too many design **reviews** that focused on the documents vs. the design (or possibly problem analysis)

Use of **unvalidated models** in simulations leading to incorrect/invalid results

- We have heard this point several times
- More importantly, how do we validate models, especially if there is an explosion of models

We focus on the goal of "25 percent reduction in time" for major air vehicle systems that must satisfy airworthiness and safety requirements required by NAVAIR. Therefore, we emphasize two key challenge areas in this subsection that have been discussed at most meetings, which include:

Growth and complexity of software and the verification challenges, which are essential to airworthiness and safety

Cross-domain model interoperability, consistency, and transformability with the required semantic precision to provide accurate information for decision making

COMPLEXITY OF SOFTWARE AND THE VERIFICATION CHALLENGE

The strict requirement for safety and airworthiness for the NAVAIR air vehicle systems requires comprehensive rigor in verification. As 90 percent of the functionality of in a 5th generation air vehicle system is in software that implies a significant amount of software verification.

One particular challenge that we discuss in the meetings with organizations is software. Jaime Guerrero, one of our NAVAIR sponsors that attended every organizational meeting, usually discusses his effort on the Joint Strike Fighter (JSF) stating that: "90 percent of the functionality in 5th generation air vehicle systems (e.g., F-35) is in software." While the first flight of the F-35 was 15-December-2006, we still do not have a flight certified system. While we have 1D, 2D, and 3D types of physics-based models for simulation, optimization and analysis, we do not have very many models of software where formal analysis and testing can be performed to the degree it is done for physics-based models.

There are reports that software testing is taking a long time (GAO report [42]). While there is use of models, the detailed software behavior is often written manually, which minimizes the ability to formalize analysis, generate the code, and automate test, with the possible exception of Simulink (but not everything is modeled like a control system). This is one of the greatest concerns to the goal of reducing 25 percent of the time.

To put this challenge into perspective, NASA presented industry data indicating that verification is 88 percent of the cost to produce DO-178B Level A[8] software, and 75 percent for Level B software [22]. These types of verification requirements are required for many aspects of NAVAIR vehicles, such as the control laws for the F-35. As shown in Figure 10, the DARPA META pre-program solicitation (META) describes how continually increasing complexity impacts the verification costs of software and delivery time [10]. META claims that the fundamental design, integration, and testing approaches have not changed since the 1960s, as shown in Figure 11. The META program goal is to significantly reduce, by approximately a factor of five, the design,

[8] DO-178B/C is the Software Considerations in Airborne Systems and Equipment Certification document dealing with the safety of software used in certain airborne systems. Level A is a characterization for the most safety-critical aspects of the software, and required a more comprehensive amount of software verification.

integration, manufacturing, and verification level of effort and time for cyber physical systems. The complexity has increased for integrated circuits, as it has for software-intensive systems, but the developers of integrated circuits have maintained a consistent level of effort for the design, integration and testing efforts, as reflected in Figure 10. The need is to understand key reasons why software-intensive systems production is different from integrated circuits. One fundamental difference is that software behavior requires nonlinear operations and constraints that are implemented on computing hardware where operations are performed and results stored in floating point representations. This makes the automated verification problem more challenging than for integrated circuits, where automated verification and analysis is based primarily on logic or bit-level manipulations. Chip developers used to rely on simulation, much like software development uses debugging and manual testing, but the chip verification would cost more than 50 percent of the effort and defects that escape to the field could cost $500M[9]. They now rely more on formal methods and tools to support development and verification.

Implications:

In the past software might not have been a major concern moving through the PDR or CDR decisions, but it may be going forward as we have not heard of many breakthroughs that can significantly reduce the time for software verification of safety-critical systems

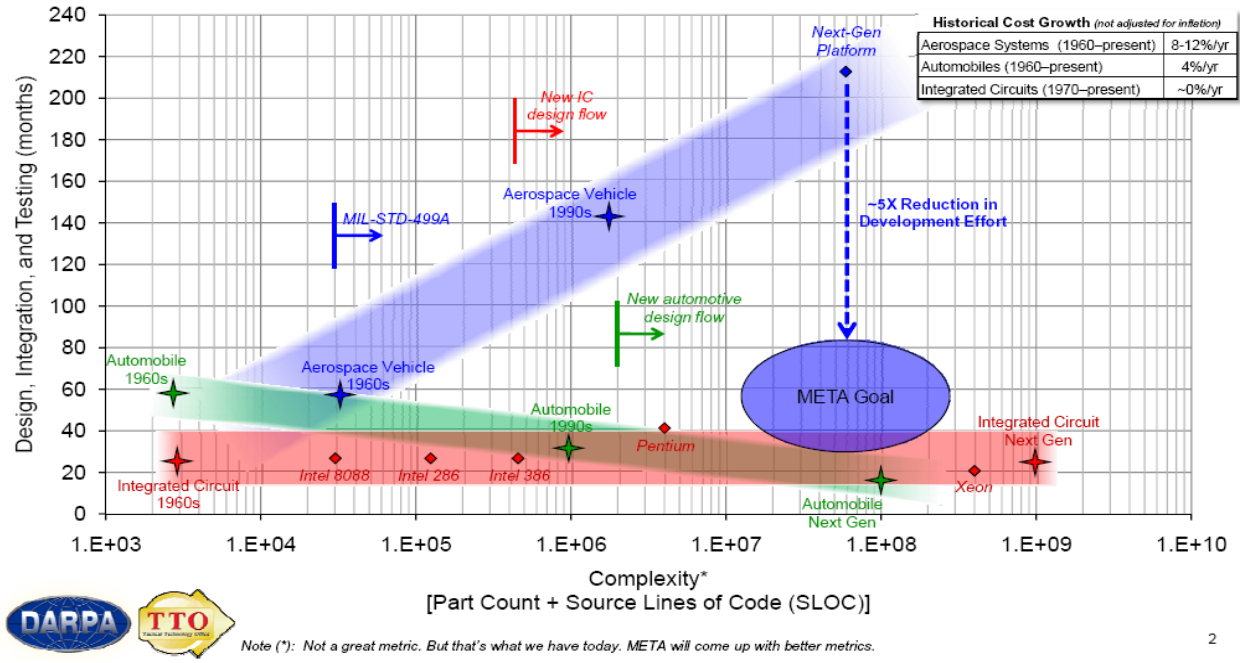


Figure 10. DARPA META Program[10]

[9] http://en.wikipedia.org/wiki/Pentium_FDIV_bug

[10] DARPA META program APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED

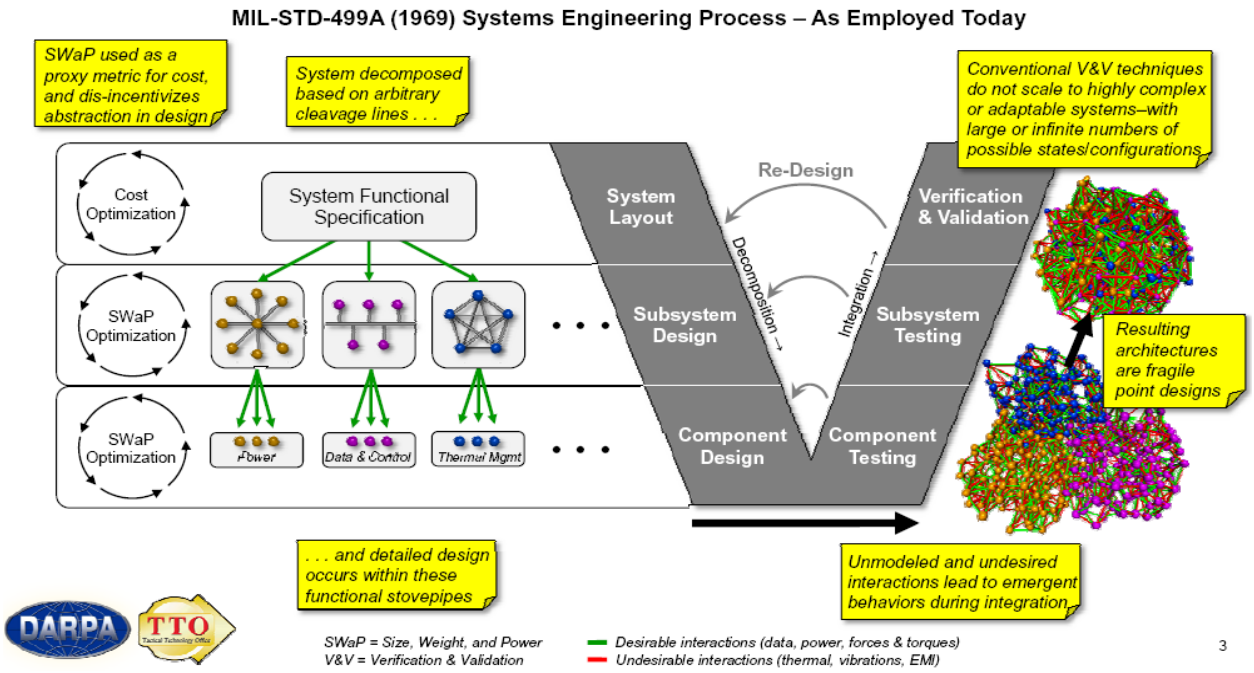


Figure 11. META Program Claims Conventional V&V Techniques do not Scale to Highly Complex Systems[11]

There may be many differences between hardware and software, and we briefly summarize the points:

Software behavior often relies on floating point variables with nonlinear relationships and constraints, which is not the case in integrated circuits

This requires different mechanisms for analysis and verification than are used in hardware analysis and verification

Other than models like Simulink, the detailed software behaviors (functions) are still written mostly by hand, limiting automated analyses

- Some discuss the use of automated generation of code
- But many are using coding frameworks, which can generate the code structure, but the detailed behavior is written in the code using languages like C++
- Newer approaches that rely on domain-specific modeling are being researched through DARPA efforts, but most have not become mainstream [17], [74].

Figure 12 was originally created in the early 2000s. We updated the chart to reflect that the number Lines of Code (LOC) in the F-35 is about 9,000,000; these are only the core, and do not include software in radars, weapons, etc. This is almost an order of magnitude beyond the F-22. We are trying to get data to make the comparison, even though we know there are many types of technology differences from the way software is produced today versus in the 1990s.

[11] DARPA META program APPROVED FOR PUBLIC RELEASE. DISTRIBUTION UNLIMITED

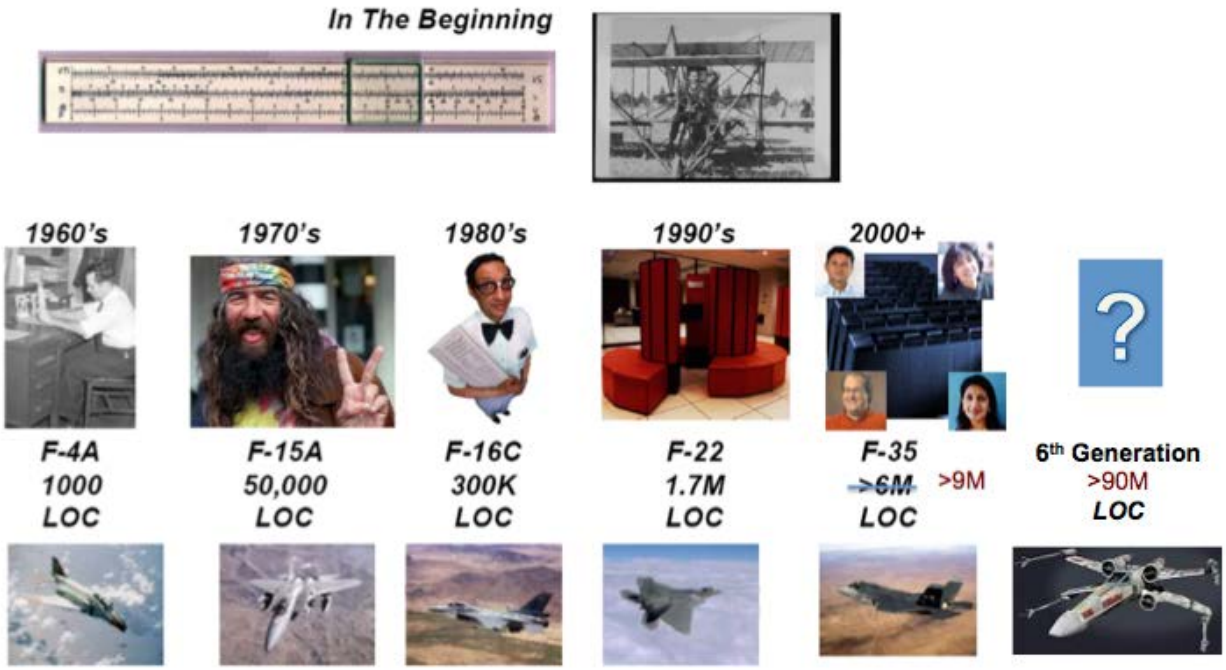


Figure 12. Augustine's Law: Trend Indicates that Software Increases by an Order of Magnitude Every 10 Years[12]

The problem is that with software there are diseconomies of scale [20] as reflected in Figure 13:

In software, the larger the system becomes, the greater the cost of each unit

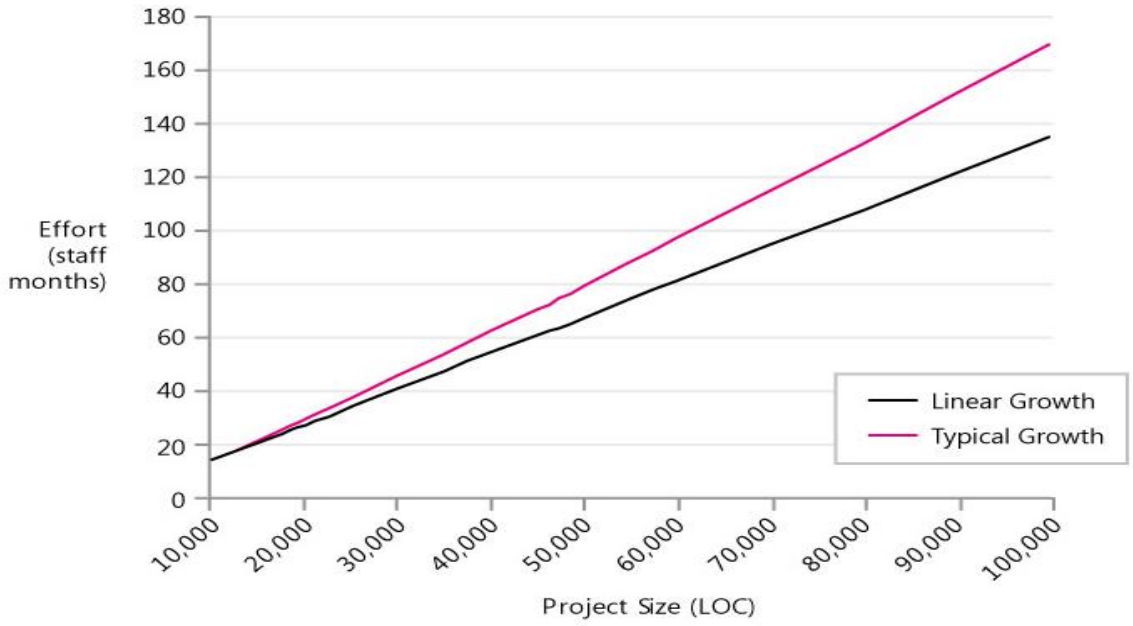
If software exhibited economies of scale, a 100,000-LOC system would be less than 10 times as costly as a 10,000-LOC system, but the opposite is almost always the case

Based on data from Boehm, as shown in Figure 13, the effect of the diseconomy of scale is not very dramatic when the range is within the 10,000 LOC to 100,000 LOC

Some automobiles may have several million lines of code, but they are distributed to small micro controllers

No one piece of code, at least today ever gets to be the size or complexity of Mission Systems Software in the F-35

[12] Image concept inspired by Ken Nidiffer as it relates to a Norm Augustine, former CEO of Lockheed Martin.



Source: Computed using data from the Cocomo II estimation model, assuming nominal diseconomy of scale (Boehm, et al 2000).

Figure 13. Complexity Results in Diseconomy of Scale In Software often impacting size, scope and cost estimates

As shown in Figure 14, we wanted to put this in context in order to discuss a key criterion of the objective, which is to reduce the current time for development by 25 percent. If "Augustine's Law" holds true, and our sponsors believe it will, then model-centric engineering would need to increase development productivity by about 13 times the rate we produce systems/software today.

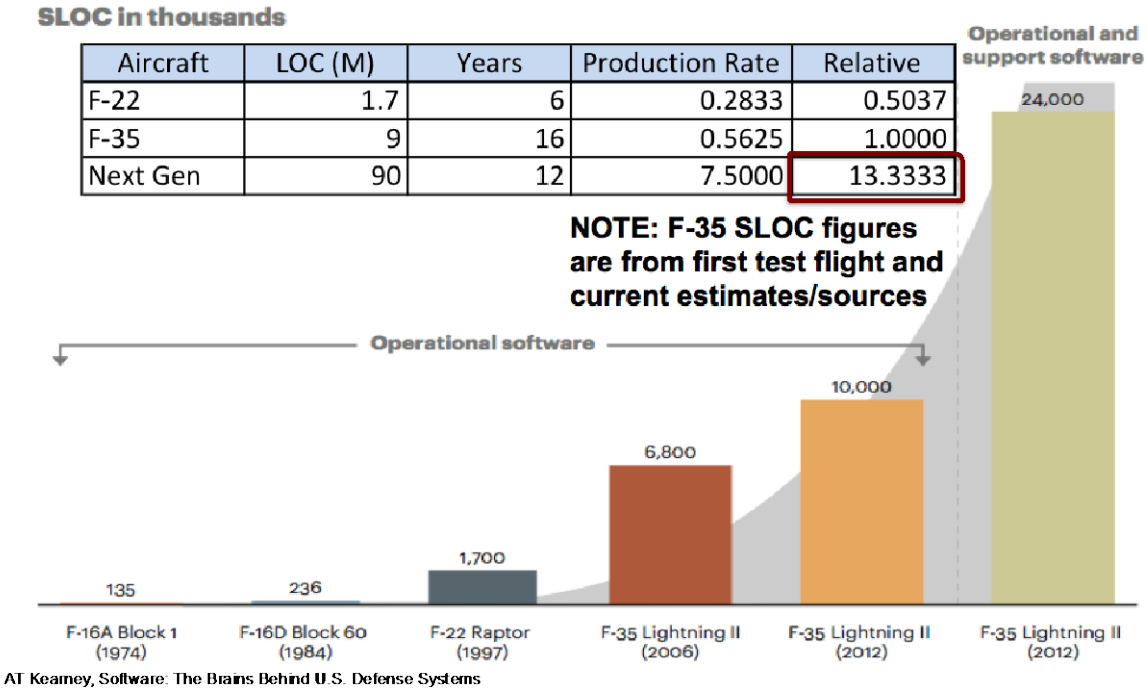


Figure 14. Number of Source Lines of Code (SLOC) has Exploded in Air Vehicle System

Recognizing that we needed to consider potential scenarios to address the issue. We identified some data sources, as we did not get much real data about many topics from the organizational discussions. We used information reflected in Figure 15 [27], a chart which are sponsors refer to often, to hypothesize some scenarios to remove defects that are introduced on the left side of the "V" in order to eliminate costly work (rework) on the left side of the "V" in order to reduce time by 25 percent. These are two possibly scenarios that can address the 25 percent reduction time:

Scenario 1:

- Increase defect removal on left side of "V" from 3.5 percent to 20 percent
- Other phases reduced **uniformly**

Scenario 2:

- Increase defect removal on left side of "V" from 3.5 percent to 24 percent
- Other phases reduced **proportionally**

Unit cost (x = 1000) actor

Factor weights from prior chart

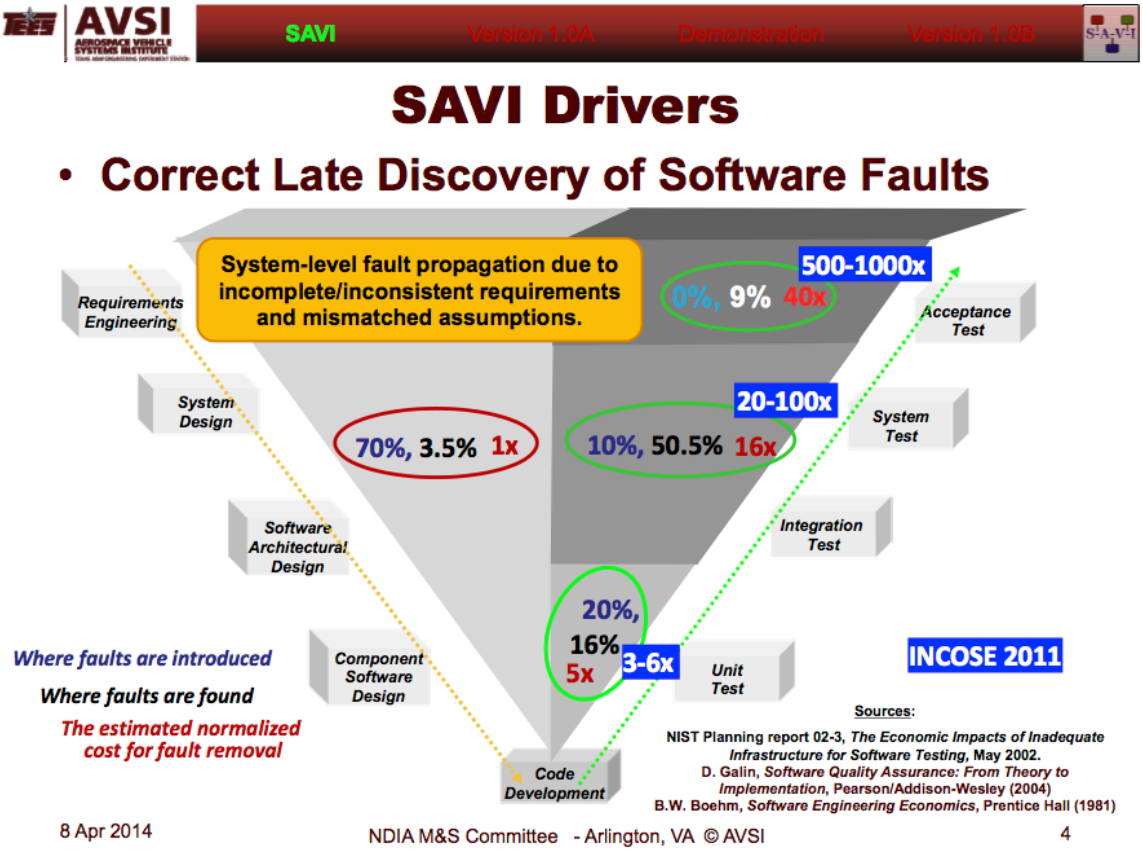
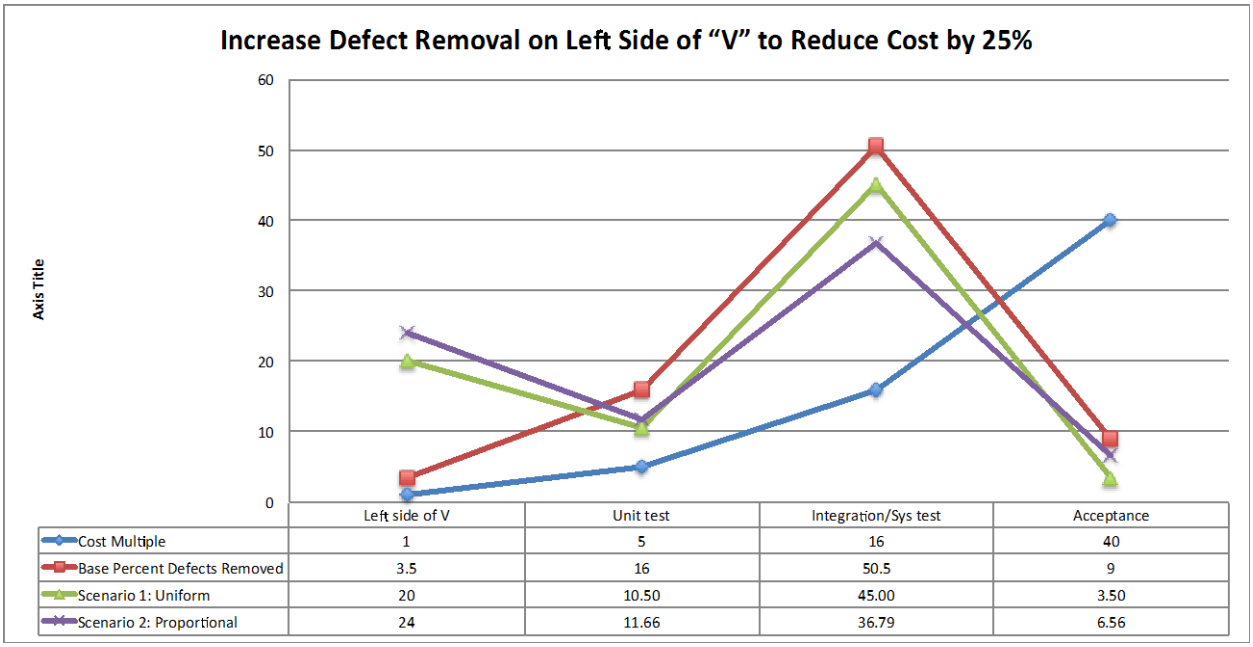


Figure 15. Perspective from SAVI on Introduction and Removal of Defects



	Left side of V	Unit test	Integration/Sys test	Acceptance
Cost Multiple	1	5	16	40
Base Percent Defects Removed	3.5	16	50.5	9
Scenario 1: Uniform	20	10.50	45.00	3.50
Scenario 2: Proportional	24	11.66	36.79	6.56

Figure 16. How Much Improvement in Defect Removal on Left Side of "V" is needed to Reduce Cost/Effort by 25 Percent

Implication:

- We need to think about this problem differently, and software does matter to NAVAIR at PDR and CDR, and JSF is a prime example
- We did not get many new answers relative to this item from our industry visits, but we know that the F-35 first flight was 15-December-2006, and the latest GAO report indicates that software testing is still a key problem [42]
- It is important to note that the mission system software has very unique capabilities (e.g., data fusion), and these types of capabilities are unlike a new Microsoft operating system that is often beta tested by millions of people prior to release; it is also not safety critical
- Our ability to meet the 25 percent reduction in time is likely to be driven by our ability to verify software, as it seem inevitable that the number of LOC will continue to increase
- We need to understand the conditions limiting the way to do the verification now, and determine if there are alternative (and possibly radically new) approaches

Finally, we had discussions with organizations that are researching the use of quantum computing focused on addressing the ever-increasing challenge of verification and validation (V&V) in systems that are increasing in complexity. They stated that V&V costs are growing at the fastest rates of any system component and rates are expected to accelerate with exponential growth in software size and system complexity driving exponential growth in certification costs. These types of technological breakthroughs can also be factored into our scenarios that model-centric engineering will change how we work, and that will reduce or eliminate some challenges.

LACK OF PRECISE SEMANTICS FOR MODEL INTEGRATION, INTEROPERABILITY, AND TRANSFORMATION

With all of the unique and advanced capabilities, systems engineering needs to manage the integration of all of the disciplines across many domains:

We may have a “sea” of models, simulators, solvers, etc., but they lack consistent meaning across or between them

There is lack of precise semantics especially in both behavior of models and timing/interactions of models

- We have covered this point many times in working session
- This point was made at the kickoff meeting, and has been reported by many others in our visits [63]

This is limiting the full spectrum of analyses and simulations needed to provide adequate coverage over a system’s capabilities

Some are looking at how to work and integrate a federation of models and digital assets, but that is not an ideal solution

We did hear information presented by some of the commercial organizations that are working on integration mechanisms such as Open Services for Lifecycle Collaboration (OSLC), which could provide better ways to get integrated views of different artifacts, some models, which would be

helpful in a new type of review process. However, the capabilities discussed are more likely to support a transition to a model-centric approach rather than a radical transformation, as many of the capabilities are oriented towards collaborative work environments, project management, and integrated views, rather than the technically advance modeling and analysis capabilities that are needed to achieve the 25 percent reduction in time.

Both industry and our NAVAIR sponsors believe that they can and are beginning to address some of these issues.

SUMMARY AND NEXT STEPS

Since the kickoff of phase 2 of this effort under RT-118, we have spent most of our time conducting and documenting the information we received from the 29 discussion meetings as reflected in Table 1. During RT-118, we held periodic (~monthly) working sessions at NAVAIR. The working sessions usually cover the status of all four tasks.

At this point we have identified challenges and gaps, but have come up with some scenarios where we can argue that it is technical feasible to meeting the 25 percent reduction in time to develop large scale air vehicle systems enabled by model-centric engineering. However, we have work to do in:

Characterizing a “radical transformation” and the associated End State

Clarify how to systematically address model validation (ensure model “integrity”)

- Term used by our sponsor, with the implied meaning that we have trust and/or evidence in the accuracy of model’s predictive capabilities

We will continue to have follow-on discussions with a number of the organizations that we have visited (Task 1)

We need to discuss the content and representation of the lexicon (Task 2) that has been transferred to NAVAIR

Sections 3 through 6 provide additional details related to tasks 1 through 4. Section 7 provides a summary and discusses a few key items that need to be discussed moving forward.

ASSESSING THE STATE-OF-THE-ART MBSE

The material in the remainder of the document has been extended or refined from the RT-48 Final Technical Report [16]. Some of this material has been documented in the bi-monthly status or working session meeting minutes.

Section 2 provided a comprehensive overview of the responses from our 29 discussion meetings with industry, government and academia. Our team developed a guideline for our collective NAVAIR team to hold discussions in an effort to understand the most state-of-the-art and holistic approaches to model-centric engineering. The objective for our team members was to facilitate conversations through discussions that draw out insights into leading advances in model-centric engineer. We agreed early on with the sponsor that open-ended discussions, as opposed to surveys, would bring out new and more innovative approaches and strategies. We were particularly interested in demonstrations of actual technical capabilities, but we only saw a few. We also wanted to understand the critical gaps and limitations too, which we comprehensively summarized in Section 2.

Status: we have completed all planned discussion. However, we are planning some follow-up discussions on some of the challenge area topics. Finally, our sponsor has requested that we try to have a similar discussion meeting with DARPA.

DISCUSSION NARRATIVES AND MEASUREMENT SUMMARY

We created a collection instrument to provide a constructive approach to conduct a discussion with organizations as well as a way to provide some type of quantitative measure associated with using subjective information to rate the “state-of-the-art” of a holistic approach to model-centric engineering. We are using a qualitative subjective approach that computes a probabilistic value associated with crosscutting factors associated with the technical Vision for this task.

The collection instrument uses an Excel spreadsheet as the input mechanism to collect factor values about an organization’s use of MBSE as discussed in Section 3.3. Each row in the spreadsheet represents the subjective information associated with one organization. The latest version of the instrument includes one organizational classifier and 22 factors.

As shown in Figure 17, the model produces two probability distributions, one for the Technical Risk State of the Art (max of 10), and another for the Technical State of the Art (max 100). We think these factor values will provide a probabilistic value that is related to the technical feasibility questions, and help in reflecting on the factors that are enablers, as well as help identify where gaps exist that must be addressed through risk identification and management. We have made some adjustments to the factor weightings based on some of the discussions we have had with organizations, and need to make further adjustments.

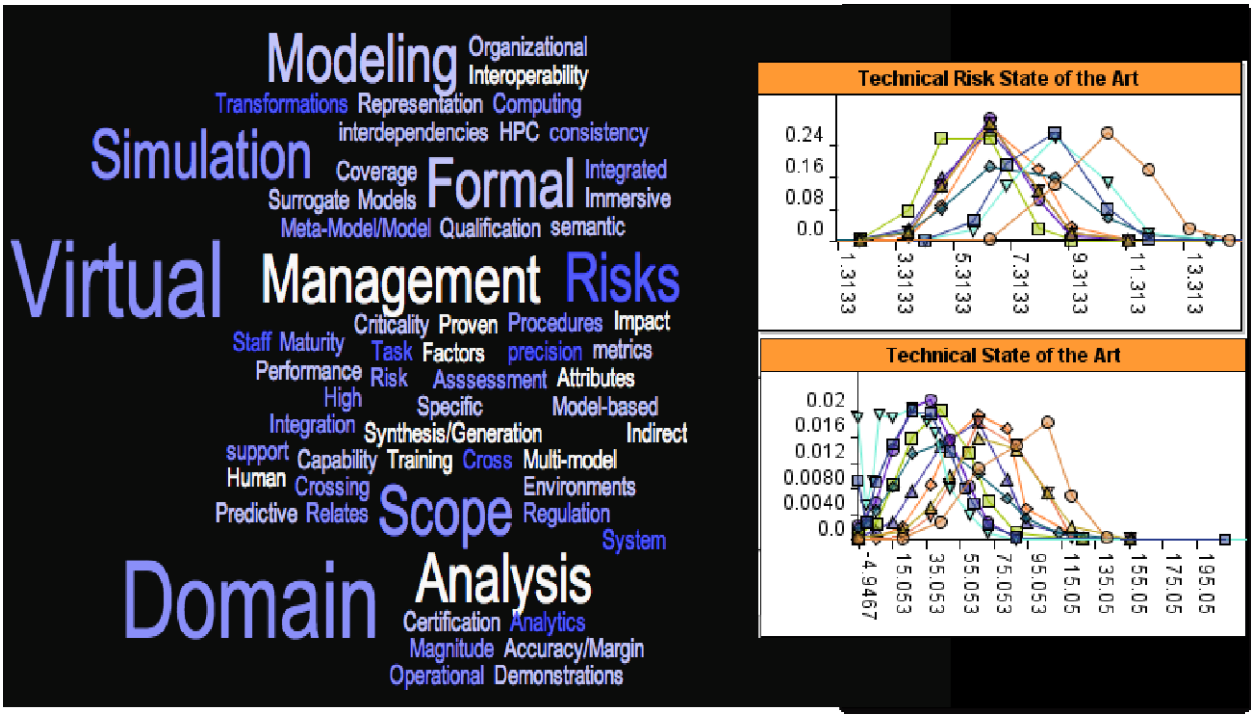


Figure 17. Measurement Collection Instrument

The analysis did highlight several of the challenge areas listed below, but in the end it was not able to deal with the software complexity issue in achieving the goal of a 25 percent reduction in the time to deliver a large scale air vehicle system. Therefore, we addressed this topic with the scenarios provided in Section 2.5.1.

Some of the enablers extracted from our discussions were (this list is not exhaustive):

Mission-level simulations that are being integrated with system simulation, digital assets & products providing a new world of services

Leaders are embracing change and adapting to use digital strategies faster than others

Modeling environments to create dynamic Operational Views (OV1) more commonly used, which used to be static pictures

1D, 2D & 3D Models have Simulation and Analysis Capabilities (mostly physics-based)

Platform-based approaches with virtual integration help automakers deliver vehicle faster

Modeling and simulation in the automotive domain is reducing the physical crash testing (e.g., from 400 to 40); this could imply that modeling and simulation can reduce test flights, which are very costly as it is difficult to get flight clearances on air craft that have advanced new capabilities Design optimization and trade study analysis

Engineering affordability analysis

Risk modeling and analysis

Pattern-based modeling based on ontologies with model transformation and analysis

Domain-specific modeling languages

Set-based design

Modeling and simulation of manufacturing

We next discussed the gaps and challenges:

Model integration, interoperability, and transformation between domains and disciplines is a challenging issue

- Still mostly stove-piped
- Systems engineering is about integration of disciplines across many domains, but there is not a lot of cross-domain integration in the simulation capabilities (only a few exceptions)
- We have a "sea" of models, simulators, solvers, etc., but we do not have consistent meaning across or between them
- Lack of **precise semantics** especially in both **behavior** of models and **timing/interactions** of models
- This limits the full spectrum of analyses and simulations needed to provide adequate coverage over a system's capabilities; it is also not well integrated "upward" into the mission simulations (although there is effort to do this)
- Some are looking at how to work and integrate a federation of models and digital assets, but that is not an ideal solution

Many believe we can "engineer" the "integration" of models/simulations to address this challenge

Increasing complexity in software, which is 90 percent of the functionality in large scale air vehicle systems

Use of un-validated models

- Note: unvalidated does not mean that the model is invalid

COLLECTION PROCESS

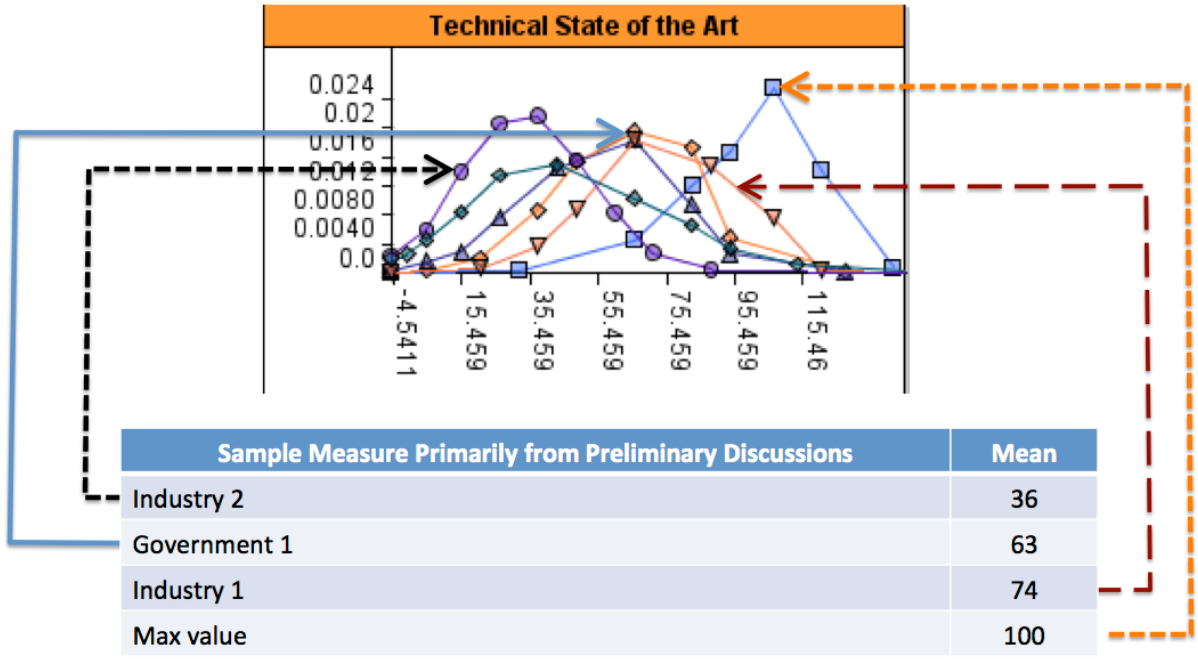
After a meeting with an organization, we request our team members who conducted the discussion to:

Complete one row of the spreadsheet; see Section 3.3 for details on the collection process

Write a short summary reflecting on the key unique capabilities of the organization

The spreadsheet responses are incorporated in a master list. The value for each factor will be entered in a modeling tool, which quantifies the subjective inputs provided to the tool, as shown Figure 18. The maximum value of the mean of the probability distribution is 100. As reflected in Figure 18, it was decided that because there are some organizations that require confidentiality or proprietary information agreements, we have decided to keep the names of all organizations anonymous. In addition, a narrative will be created for each organization; this narrative will highlight the most key capabilities and challenges, but be generalized to ensure each

organization's anonymity. Additional details about interpreting the results are provided in Section 3.3.



Sample Measure Primarily from Preliminary Discussions	Mean
Industry 2	36
Government 1	63
Industry 1	74
Max value	100

Figure 18. Collection Instrument Results

SCENARIO COLLECTION

After each discussion we complete the spreadsheet collection mechanism as shown in Figure 19 by working through the row and uses the pull down menus to select a factor value of Low, Medium, or High (see Section 3.5.2 for details on Ranked factor values). A complete list of factors is provided in a worksheet tab of the spreadsheet collection mechanism titled: Factor Meaning-Definition. Example definitions are provided in Section 3.3.3, with some additional rationale; a complete set of definitions is provided in Discussion Collection Instrument Guide and provided in the back up material of this report.

Team members may want to use one spreadsheet to collect all of the discussions; it is possible and acceptable that after a few meetings with organizations that one or more of the factor values be changed in order to be more globally consistent. The key is not to identify a particular organization, rather the objective is to identify if there are state-of-the-art methods, tools, processes and innovative strategies that are being used to significantly advance the development and deployment of systems through model-centric engineering and related approaches, and to incorporate these concepts in the Vision model (see Section 5).

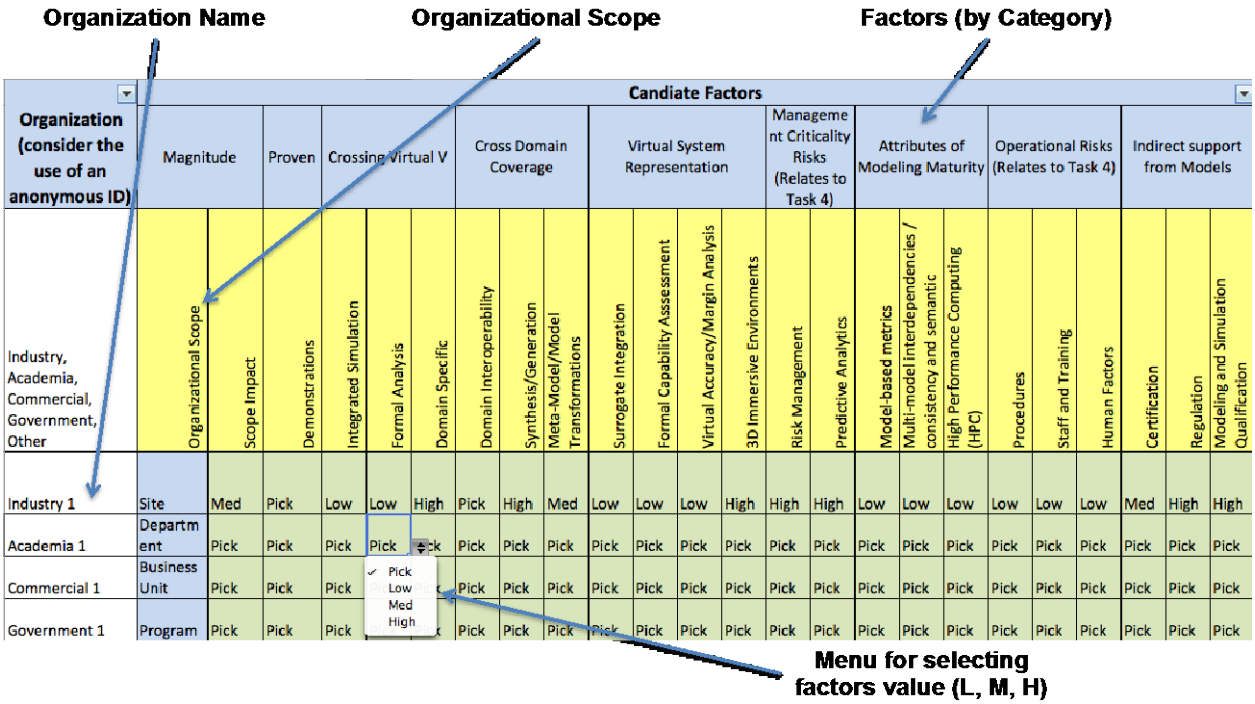


Figure 19. Spreadsheet Instrument Collection

ORGANIZATIONAL TYPE

The general convention used is:

Academia – this should include strictly academic organizations; other organizations performing research should either be Industry, Commercial or Government

Industry – these are organizations that are using MBSE to develop products and systems (e.g., those contractors to NAVAIR that produce air vehicle systems)

Commercial – this is a special case of Industry that relates to MBSE product developers

- These organizations either develop MBSE tools and services, or may apply them with Industry or Government customers
- These organizations are in the list, because they may have insights into some of the leading or novel uses of the tools, and they are aware of the need to continually advance their own product and services

Government – this includes military, and other non-military organizations such as Department of Transportation, and the FAA

ORGANIZATIONAL SCOPE

One challenge for some of the initial uses of the collection mechanism was to appropriately reflect on the organization scope for which these model-centric engineering usage questions apply. Remembering that the key objective of the survey is to assess the "Technical Feasibility" of "doing everything with model." We recognize that actual adoption can be difficult, and might

not make sense on older systems. Therefore it is probably best to hold discussions with individuals in the roles such Chief Engineer, Chief Technical Offer, Program Manager or some MBSE technical experts in the organization. To carry this a step further, it might also be important to keep the "systems" perspective in mind, because some of the concepts discussed may have been applied at the hardware level and possibly in types of software (e.g., the control laws for the F35 are built in Simulink, with auto code generation, and a significant portion of auto test generation), but these types of approaches may only be emerging in use at the systems level. We seek to understand how comprehensive the use, and also **need to understand the technical gaps**. The technical gaps areas will likely need to have additional focus related to risk management (Task 4).

Finally, this research is not limited to NAVAIR, however when thinking about NAVAIR systems the scope is often quite large and involves large-scale multi-year programs, where the systems are actually built by one or more contractors.

Therefore, we would like to know the organizational scope associated with the MBSE discussion: Program, Project, an entire Business Unit, Platform (e.g., a type of a specific aircraft, tank, automobile), Department, or Site.

FACTORS DEFINITION EXAMPLE

The factor categories do not necessarily relate to specific MBSE practices, rather they are higher-level characteristics of the organization's ability to leverage the use of models and the associated technologies that enable simulations and accelerate the analyses, design, synthesis, V&V and manufacturing processes. For example:

Crossing the Virtual V is a high-level category that has significant weighting in the model, because our sponsor emphasized this as a critical need and the ability to understand the design capabilities through early V&V activities at the system and mission level (as opposed to the subsystem or component level). Therefore, this factor category has three main factor characteristics:

Simulation of Integration

- If an organization has simulations of integration or integrated simulations across domains of the system, and especially at the "higher" levels of the "V," this is a likely indicator that such an organization is likely to have the ability to understand simulations of the system within the context of a mission, and there is a better understanding of the integration impacts, because the simulations are integrated or represent integration, including critical temporal aspects in simulation
- This includes the integration of surrogates, use of instrumented systems, actual system components, new prototypes, and/or in development
- Other attributes of this type of simulation, would be human-in-the-loop, as well as multi-level mixed fidelity simulations that provide the right abstractions at the right level

Formal analysis

- This means that the analysis is automated, because the models are semantically rich; we are looking for automated analysis, rather than looking at humans performing the analysis
- Models are increasingly have more semantic richness that enable automated-types of analysis, and models are increasingly being integrated (see factor category **Cross Domain Coverage)**

Domain specific

- These types of systems involve the integration of many disciplines
- Models need to provide the relevant abstractions that are related to the domain of the engineer performing the work; domain-specific modeling is an emerging type of modeling that often provides the relevant abstractions, with the semantic richness to enable automated analysis, simulation, synthesis (generation) and automated test
- DARPA-sponsored research that demonstrated the capability for continuously evolving Domain Specific Modeling and analyses in 2008 as an emerging capability and theme [31], [74]. In contrast, modeling languages like System Modeling Language (SysML) are general purpose [51] they generally lack the semantic richness needed for formal analysis leveraging for example formal methods of automated V&V [16]; while they may be understood by system engineers, control system engineers would prefer Matlab/Simulink, and other engineers may require other domain-specific models and tools (e.g., computational fluid dynamics, radio-frequency, heat transfer). However, SysML does provide an underlying framework for holding system model information [90], yet the models are not executable even with existing plug-in authoring tools [25].

DISCUSSION SUMMARIES

There are detailed meeting notes that were shared with the NAVAIR research, but they were not generally released as many of the discussions with industry and commercial organizations were governed by some type of Proprietary Information Agreements (PIA) or Non-disclosure Agreements (NDA).

Section 2 provides a summary in the aggregate of the information that will be made publically available.

PREDICTIVE MODEL

This section is provided for those interested in more details about the mechanism for converting the subjective factors into a quantitative number. The model is created using a Bayesian Network [73] (BN) tool. There are two basic reasons we selected this approach, BNs:

Provide for the translation of subjective information into quantitative probabilities

Allows for the use of subjective expert qualitative judgment and captures the casual relationship between subjective factors

The outputs are also probability distributions, which means that they provide some type of range to provide a comparison between different organizations. The specific numbers are not necessarily as important as our ability to compare different organizations and relate the responses back to advanced uses of MBSE and related enabling technologies. While no organization may have all "High" values, this approach provides a way to look at relative comparison in conjunction with the narratives. Each of the nodes in the BN shown in Figure 20 provides a type of weight called a conditional probability. We have used the team's judgment to weight the different nodes in a way that would relate to evaluating the key question for this task: is it technically feasible to "do everything with model." In addition, we will refine the weightings as we proceed through discussions.

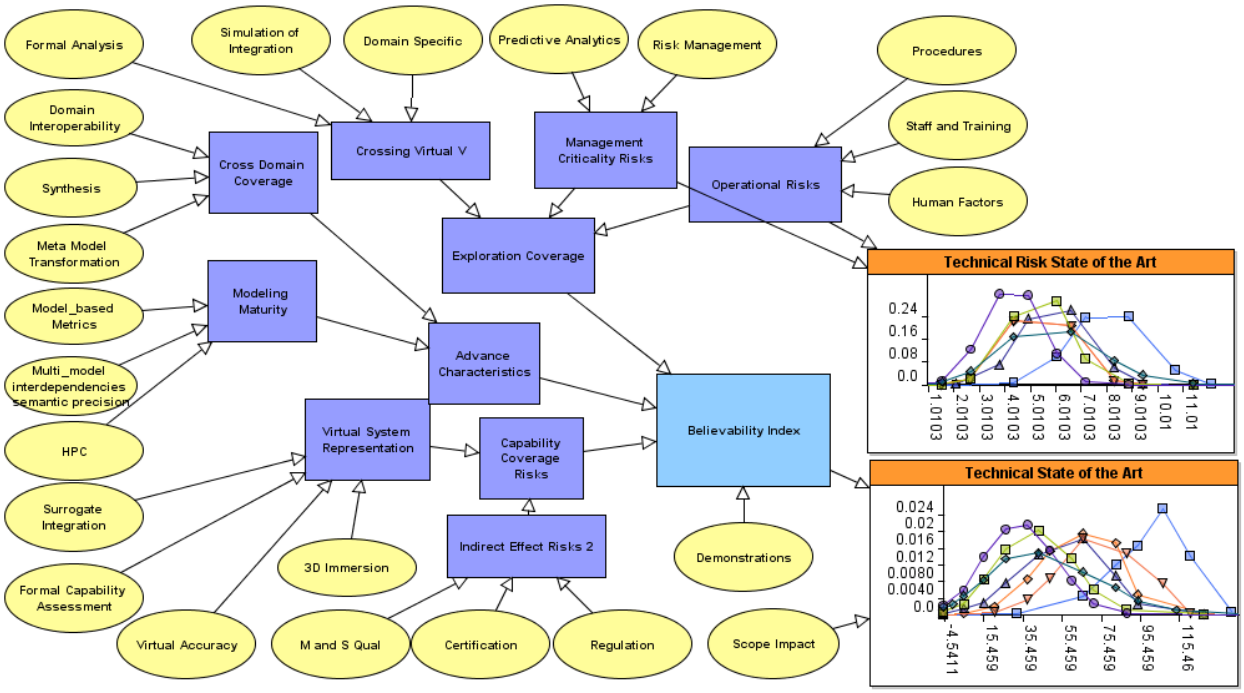


Figure 20. Bayesian Network Underlying Collection Instrument

RATIONALE FOR BAYESIAN NETWORKS

A Bayesian network is a representation, which organizes one's knowledge about a particular situation into a coherent whole [32]. They are increasingly being used in the modeling of uncertain and incomplete knowledge. Bayesian thinking is inherently more intuitive than many other evaluation techniques; it best reflects commonsense thinking about uncertainty that humans have. We frequently use words like "likely," "rarely," and "always" to express varying degrees of uncertainty. Subjective probability is our way of assigning numbers (between 0 and 1) to these different degrees of uncertainty, and our probabilities can change as we are presented with new information, or we have new experiences which cause a shift in beliefs or expectations. When this shift occurs, the way our probabilities change are governed by Bayes' rule.

A Bayesian network, as used in this framework, is a joint probability distribution and as such, any question that can be asked in a probabilistic form can be answered with a stated level of confidence. Some typical questions might be:

Given a set of effects, what are the causes?

How can an outcome be controlled, given a set of circumstantial values?

If we model a causal relationship, what result would an intervention or change bring?

While there are several ways to structure a Bayesian network, we used prior experience to structure the model. The subjective factors in the spreadsheet instrument map directly to the yellow oval nodes of the BN model. The purple rectangles are intermediate nodes and generally relate to factor categories. The orange rectangles represent the probability outputs of both Technical state of the art (Task 3) and the Technical Risk state of the art (Task 4).

DATA - LIKERT SCALES (RANKED SCALES)

The subjective factors in the model use a Ranked node type, which is a type of Likert Scale. It is important to note that although Likert scales are arbitrary, they can retain a level of reliability for our use. The value assigned to a Likert item has no objective numerical basis, either in terms of measure theory or scale (from which a distance metric can be determined). In this case, the value assigned to a Likert item has been determined by the researcher constructing the Bayesian network, but can be refined as the research progresses. The results have been a balanced representation of strata and detail.

Typically, Likert items tend to be assigned progressive positive integer values. Likert scales usually range from 2 to 10 – with 5 or 7 being the most common. In this model, 3 levels are used, at least for now as it minimizes the number of computational states, which minimizes time for the analysis. The progressive structure of a Likert scale is such that each successive Likert item is treated as indicating a 'better' response than the preceding value. Note that the direction of 'better' (i.e., Higher) depends on the wording of the factor definition, which is provided in Section 3.3.3.

In terms of good practice, a bias in the computations may result if the suppliers of data for the framework do not agree on the relative values of each factor. However, there are enough factors that a bias in a one or two values will likely not skew the results significantly.

COMMON MODEL LEXICON

The team was tasked at the kickoff meeting to create a common lexicon for things related to modeling in the systems engineering domain, and in fact, in the broader engineering space. An example of this is what is meant by the word "model." Most engineers will agree that a model is a facsimile of reality. Yet, to an industrial engineer, a model may represent a production facility; to a mechanical engineer it may be a finite element model analysis; to a systems engineer it may be an IDEF0 [48] or a SysML representation of the system, subsystem, or some lower level element. None of those perspectives are wrong; they are just different views of some part of the same enterprise.

Some claim that there is no existing model lexicon or taxonomy [7], although there are a number of different types of taxonomies that all fit within the more general context of a model lexicon [29], [90]. The Object Management Group (OMG) in conjunction with INCOSE has established an Ontology Action Team to work on similar efforts [68]. The NDIA Modeling & Simulation Committee is about to approve the Final Report on the Identification of Modeling and Simulation Capabilities by Acquisition Life Cycle Phase [8].

> Status: we have captured over 300 named lexicon items related to the term “model,” including levels, types, uses, representations, standards, etc. The details are described in Section 4; we have delivered these model-lexicon artifacts to NAVAIR for them to post internally.

ONTOLOGY VS. LEXICON

According to Wikipedia, ontologies are the structural frameworks for organizing information and are used in artificial intelligence, the Semantic Web, systems engineering, software engineering, biomedical informatics, library science, enterprise bookmarking, and information architecture as a form of knowledge representation about the world or some part of it [86]. The creation of domain ontologies is also fundamental to the definition and use of an enterprise architecture framework.

A lexicon is a similar concept – it is normally a book or glossary like document, or words (and their definitions) in a language or domain, arranged in alphabetical order. The team decided that a simple glossary would not be sufficient because it does not show the relationships between terms.

In simplistic terms, an ontology becomes a complex network of words, and their relationships to each other. A lexicon is a glossary. Neither was exactly what was needed for this project. Instead a hybrid is needed. The team needs something that provides definitions and simple relationships – not complex, rigid definitions. We chose to use the word Lexicon, though the words could also be represented in a tree-like structure that is common for ontologies.

TOOL FOR REPRESENTING WORD RELATIONSHIPS

There are tools available for creating ontologies. There actually exists a class of workers that consider themselves Ontologists. These tools come in many different flavors – from open source tools to commercial tools. The common thread is that they create graphical representations as shown in an example in Figure 21. These tools require rather rigorous definitions and relationships to complete. The open source tools are actually very good, and very robust. However, after some evaluation of available open source tools, the team decided that it would be better to create a straightforward spreadsheet of terms (e.g. a Lexicon), and then create a script that could represent that lexicon graphically.

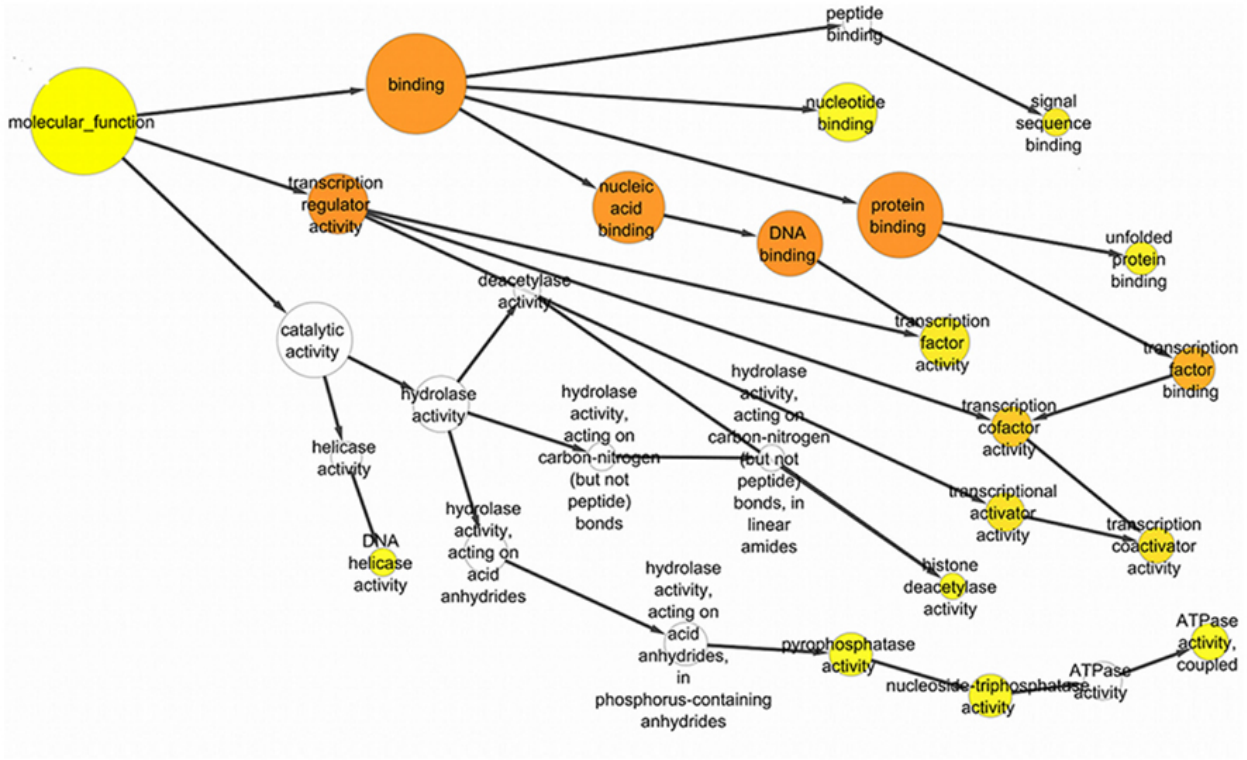


Figure 21. Sample Graphic Representation from Ontological Software

THE LEXICON

A spreadsheet was first created in Excel. At first, the team was simply capturing the words, their definition, and where it made sense, a key reference or two for that definition. Table 2 shows the implementation of this data gathering spreadsheet. Once the decision was made to create a tool to make this information available graphically, and also on the web, it became apparent that a "relationship" data element was necessary. Therefore, the data collection tool captures:

Name

Has Parents [0 or more] separate with ";" if more than one

Definition

Sample Usage

Also Known As

Key URL (optional)

The current spreadsheet represents a continuous accumulation of relevant terms, their definitions, and their classification. The initial definitions have been drawn from readily available sources on the Internet (often from Wikipedia where the assumption is that it has been created by a group of people with both knowledge and passion about the subject). In other cases members of the research team have authored a definition based on their understanding of the term in a relevant context. The team is using the spreadsheet feature of GoogleDocs to foster a collaborative effort.

Table 2. Initial Lexicon Capture Tool

Term	Definition	References
MBSE	Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification, and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases.	(INCOSE, Systems Engineering Vision 2020, Version 2.03, TP-2004-004-02, September 2007)
MBSD	An engineering approach that promotes the use of models to develop, integrate and monitor systems across disciplines, environments and scenarios.	
MBE	Model Based Engineering - Model-Based Engineering (MBE) as an approach to engineering that uses models as an integral part of the technical baseline that includes the requirements, analysis, design, implementation, and verification of a capability, system, and/or product throughout the acquisition life cycle.	(NDIA – Final Report of the MBE Subcommittee)
Concurrent Engineering	reducing time between project launch and delivery by operating elements of the process in parallel. Concurrent engineering is simply a practice of doing as much of the system life cycle design and development in parallel as possible with the inevitable rework associated with the risk of parallel development - as you get into more detail that will reveal findings that ripple back across the 'already' completed work since you are trying to tackle everything at once	

Intuitively, many of the terms in this spreadsheet are ambiguous and their meaning is highly dependent on the context and usage domain. This has been found to be true in reality also as terms are collected from various domains. It is therefore important to emphasize that this is an evolving process.

SOURCES OF INFORMATION

There were a number of sources used for this initial Lexicon. Journal papers on MBSE provided a good first cut. Interestingly, an article from the Journal of Object Technology [40] proved to be very useful. Other sources included The Open Group, the Object Management Group, INCOSE, NDIA, and Wikipedia.

WEB PRESENTATION

A short script was created that takes the information contained in the data-entry spreadsheet, and publish the results to the web. Figure 2 shows the published page as it looks at the time of this report[13]. This page includes four sections:

Model Lexicon Overview (Figure 22)

Model Representation/Lexicon (Figure 24)

- This is a generated image produced by vizGraph, but with over 300 lexicon items it is difficult to use, although it reflects the interrelationships of the lexicon elements

Hyperlinked Tree of the Model Lexicon (Figure 23)

- As an alternative, a collapsible and expandable tree (outline) allows people to understand the hierarchy of model lexicon with hyperlinks to a particular lexicon definition.

Definitions - A common structure is used for each term (Figure 25)

[13] The final location of the lexicon may move to another location.

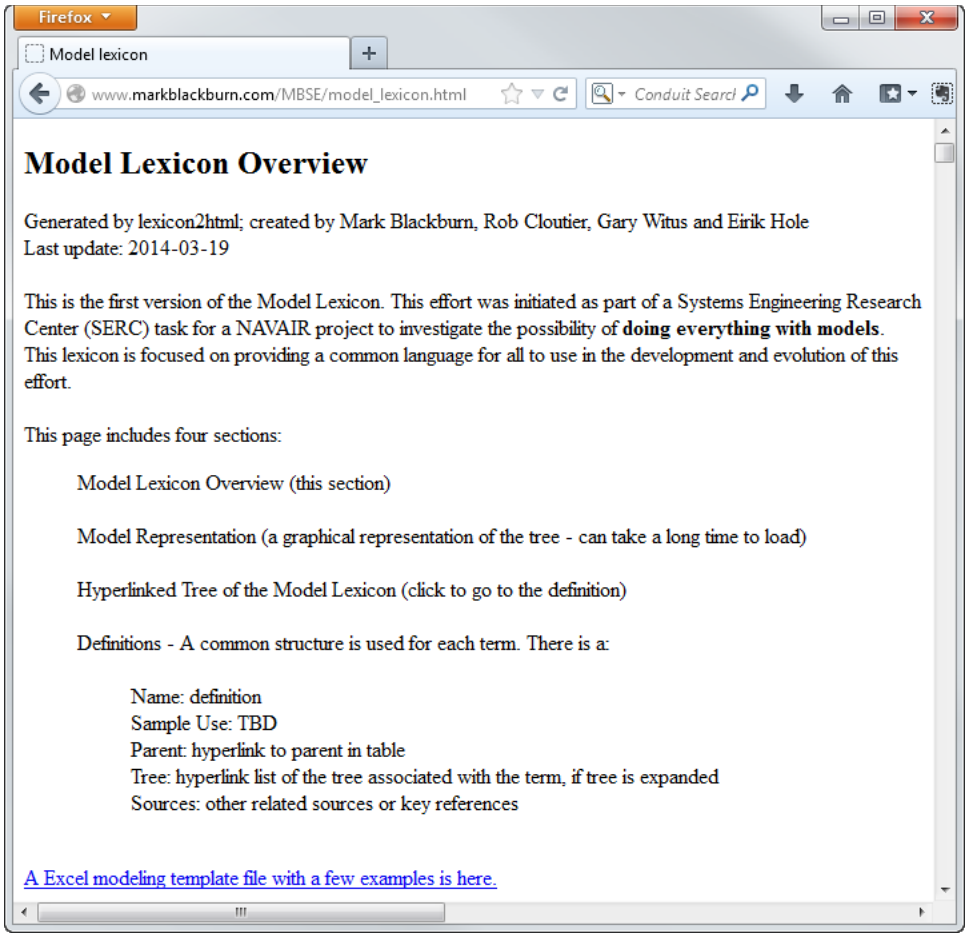


Figure 22. Published Web Page from Data Collection Spreadsheet

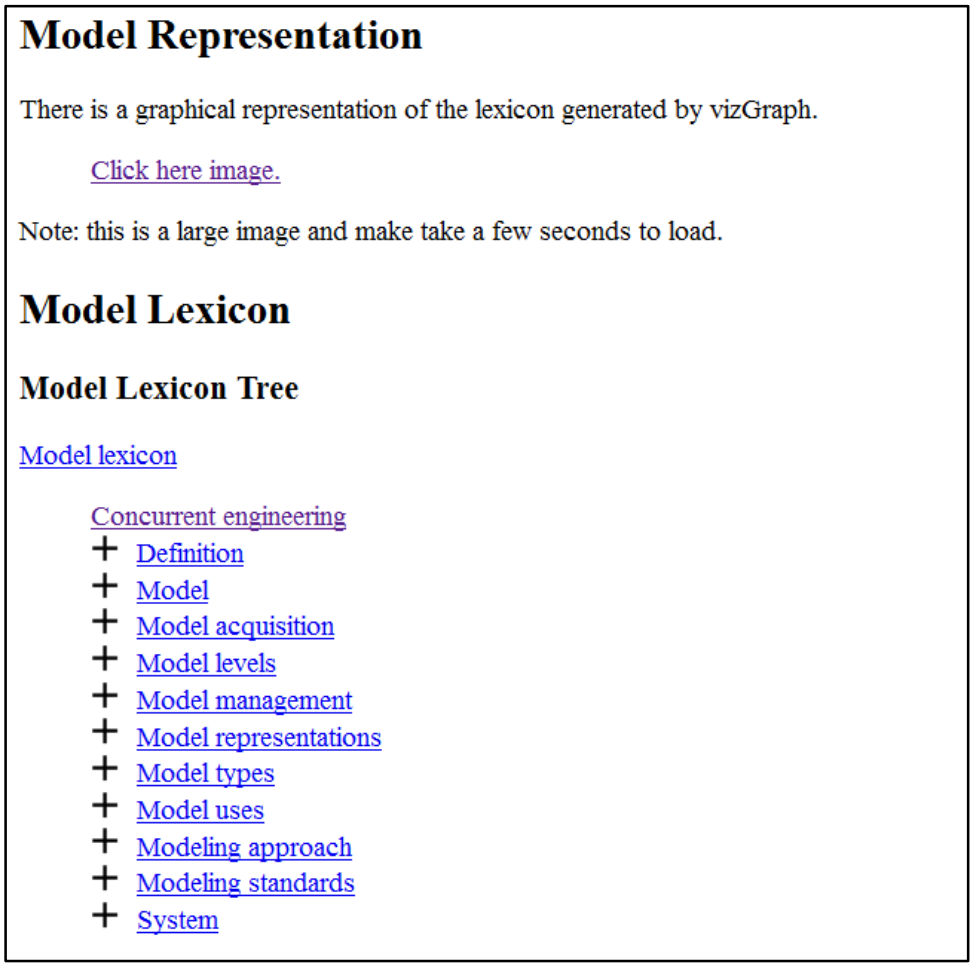


Figure 23. Model Representation and Lexicon Tree

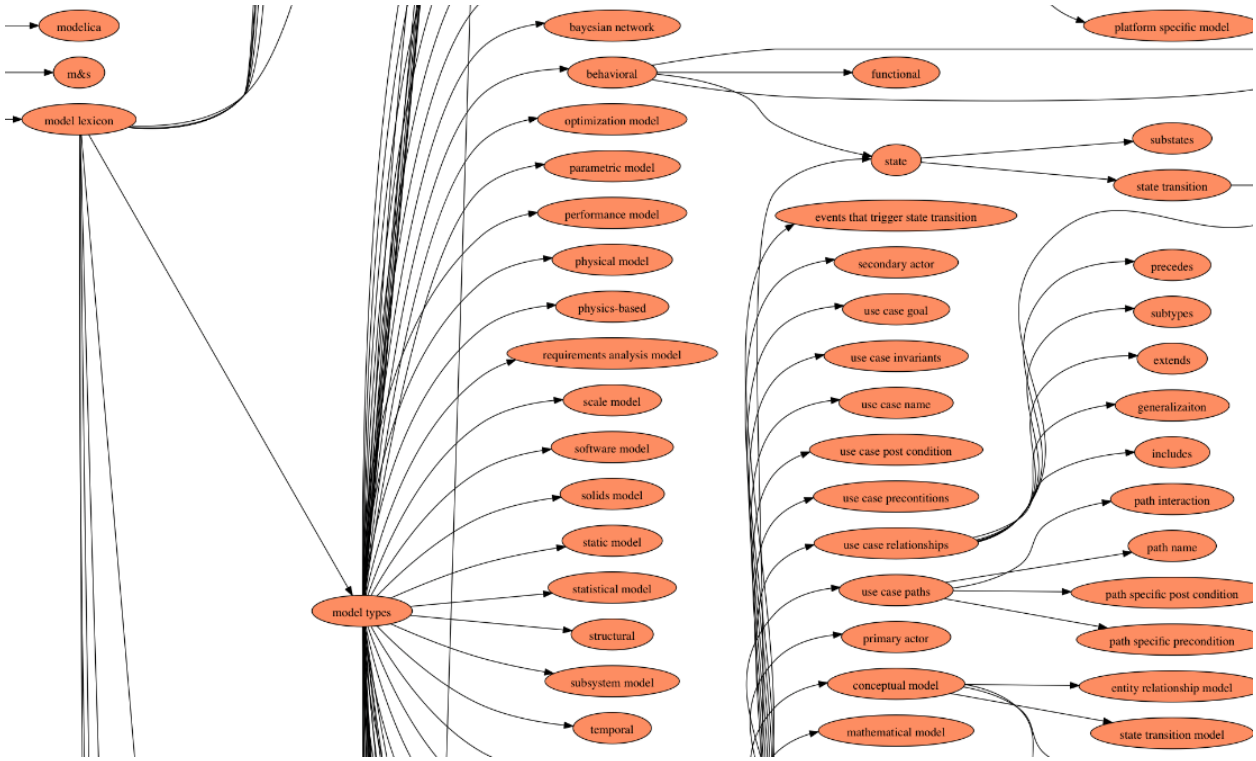


Figure 24. Partial Graphical Representation of Lexicon

The definitions table shown in Figure 25, is a screen image from the website, and includes the following columns:

Name

Definition

Parent

- This is a hyperlink to the parent in the table

Tree

- This is a hyperlink back to the collapsible and expandable tree (outline); clicking on this hyperlink takes the focus back to the name in the tree only if the item is expanded in the tree

Sample Use

Key Source (if applicable)

Model Lexicon Definitions

Name	Definition	Parent	Tree	Sample Use	Source
2d modeling:	a geometric model of an object as two-dimensional figure, usually on the Euclidean or Cartesian plane	mechanical modeling	mechanical modeling, 2d modeling		
3d solid model:	the product of 3D solid modeling	mechanical modeling	mechanical modeling, 3d solid model		
3d solid modeling:	is the process of developing a mathematical representation of any three-dimensional surface of object (either inanimate or living) via specialized software	mechanical modeling	mechanical modeling, 3d solid modeling		
Model driven architecture:	a software design approach for the development of software systems	model technique	model technique, model driven architecture		http://www.omg.org/mda
Model levels:	Level of the system or system of systems; Also discussed in terms of Resolution level. The amount of detail or degree of aggregation employed in the model or simulation	model lexicon	model lexicon, model levels	Often used in modeling and simulation world to discuss the differen types of models	
Model lexicon:	the words used in a language or by a person or group of people		model lexicon	This is a lexicon associated with terms derived from models and modeling	
Model management:	Approaches to managing models	model lexicon	model lexicon, model management	Configuration control, PLM, CATIA	

More Sample

Figure 25. Tabular Representation of Lexicon

RECOMMENDATIONS MOVING FORWARD

1. Review by NAVAIR.
2. We expect as the effort continues, team members will continue to collaborate in the definition and classification, causing discussion related to their relevance and "correctness."
3. Additionally, the intent is that the broader community will contribute examples and sample usages of the terms to improve the understanding and proper use in various contexts.
4. We will therefore provide mechanisms that allow for feedback/annotation from the community and a basic change control process.
5. It might be good to add a "comment" link on each table row on the website that could link directly to the corresponding row in the Google spreadsheet to enable the submission of a new terms and definitions directly into the spreadsheet (or database).
6. A longer-term plan would be to drive the graphical image, and textual listing from a database instead of a spreadsheet.

MODELING THE VISION AND RELATING TO THE "AS IS" AND AIRWORTHINESS PROCESS

Section 2.3 briefly discussed the concept of the Vision model, which is not a representation of a NAVAIR air vehicle system, rather the Vision model must include the required information (data) and embedded knowledge that is normally captured in documents, drawing, specifications, pseudo-formal models, and tests (some refer to it as the "total" system model [91]). This concept was discussed in terms of the containing system [1], designing system and ultimately the system instance that is the "design." This includes or subsumes every piece of information that relates to the artifacts captured in the "As Is" process, but should also include formalized information such as the inputs and outputs of modeling and simulations, analyses, surrogates, risk information, etc. and include specific versions of each tool, simulation, and analysis engine used to support the necessary evidence required to produce an airworthy system version. Ideally, this should include every piece of information to the Bill of Material (BOM), including models to manufacturing and models to training.

While it is uncertain if this concept is actually possible, it reflects on what we believe the sponsor means by the "end state" for NAVAIR. However, a truly "end state" is probably a misnomer, as this vision concept would continue to evolve. Our industry visits are suggesting that model-centric technologies are enablers for more automation and efficiencies, however we still need to better characterize how NAVAIR can achieve a radical transformation. One key discussion topic that has now surfaced in light of theme and trends from organizational discussions is the need for a "**radical transformation;**" our sponsor in the original kickoff briefing stated:

> "Blow up" the current "Newtonian" approach and move to a "Quantum" approach that recognizes and capitalizes on current and emerging trends and enabling technologies

This point was related to how we need to change the monolithic review processes. The notion of a document-centric environment could be "blown-up," because all information could be viewed in any form: native modeling representation, web-based (document-like), automatically generated into a document-like perspective that would suit a stakeholder's needed view, which could include the new SETR Manager. We have been given evidence that this too is already an existing capability [35].

Preliminary discussions with organizations suggest that some individuals and organizations understand the Vision model concept. Some are attempting to develop variants of the concept that are more specific to product development. Some have cross-business/discipline projects established to refine strategies to roll out and support adoption by programs in these different business units. Other efforts are focused more at the software level (using the characterization Model Driven Engineering [MDE]) [47]. One study cited a multi-level, multi-domain instance case that started at the campaign level moving down through the mission, engagement, and engineering levels [3]. There are also organizations that claim to be applying MBSE, yet they have not seen the benefits; we understand that there are often adoption challenges [18], and that is why our sponsor has directed us to focus on the technical feasibility for this phase of the research.

The following subsections present additional details beyond that provided in Section 2 that is related to the research investigation, working sessions, and task scoping and refinement as it has evolved during this phase of RT-118. This sections is organized as follows:

Summarize the team involvement on the outputs of Task 3

Provide an information containment and operational perspective on a Vision concept

Scoping the boundaries and interfaces between the Program of Record and Mission Analysis

State of the "As Is" Process

State of the Airworthiness Process

Perspectives on how to model the Vision

Straw Man Vision Concept

Model-centric engineering perspectives derived from research and discussions

- For example, how model-centric tool changes subsume process
- Clarification through an example of reference model (or reference architecture)

CONTEXT FOR RELATING "AS IS" AND AIRWORTHINESS ARTIFACTS AND PROCESS TO VISION

From a high-level perspective, as reflected in Figure 26, Task 3 is a collaborative effort being worked by our SERC team, SMEs from NAVAIR, Naval Post Graduate School (NPS), MITRE, SOLUTE, and consultants who have extensive NAVAIR and aircraft system engineering experience. This section provides a summary of the efforts. The following enumerates subtasks for Task 3 (the list order is aligned with the elements in Figure 26):

1. The NPS team is developing a CORE[14] model representation of a derived list of artifacts that are currently produced to support NAVAIR System Engineering Technical Review (SETR) process
   - It is important to understand the artifacts that are produced to comply with the "As Is" process, along with the relationship and dependencies among these artifacts
   - In the Vision, the information described in these artifacts (some of which are models today) must be ultimately represented in models (digital form), or be derivable from models
2. Representation of the "As Is" process, which relates to the DoD 5000.02 and SETR process
   - The analysis of the "As Is" artifacts and process should provide a means to assess the completeness of the Vision, and help people understand how a process would work when transitioning from a document-centric operational model to a model-centric approach
   - As we are attempted to leverage existing efforts, we looked at the Acquisition Guidance Model (AGM) developed by MITRE [30], but this did not have another of the NAVAIR specific information

[14] We are not promoting any specific modeling tool, and have discussed moving to a more dynamic modeling approach such as Simulink, but the model is still currently in CORE.

3. The MITRE and SOLUTE team partners are developing a representation of the Airworthiness Process[15]
    - This effort will characterize those critical aspects that make the NAVAIR requirements more challenging than for other organizations
    - The types of required Airworthiness evidence (e.g., Flight clearance) must be identified and presented either in some model representation (4) and/or support risk-based decision making, which should be captured in conjunction with the Vision (5)
4. SERC collaborators are developing the model of the Vision ("end state") representation that subsumes all information that is currently represented in the "As Is" process, if deemed to be necessary, and all of the associated digitized automation that is required to transform the process
5. SERC collaborators are developing the integrated risk framework (see Section 6 for details)
6. The associated process for applying the Vision model; in many instances, when the information is formalized in model, a corresponding model-driven automated workflow is also automated, however, because of the aspects of risk and airworthiness, it is likely that there are some human-driven steps in the process
    - See Section 5.8.1 for an example that describes how a modeling tool chain can subsume several process steps normally performed manually in a non-model-centric environment [13]

Implication: As part of a radical transformation, we have hypothesized that if every piece of information could be captured digitally that all of the information would be digitally linked too, and this would completely subsume the process (there would be no process in a radical transformation)

[15] This effort was started in our February working session and is being supported by our MITRE team partner and SOLUTE is now involved in this effort.

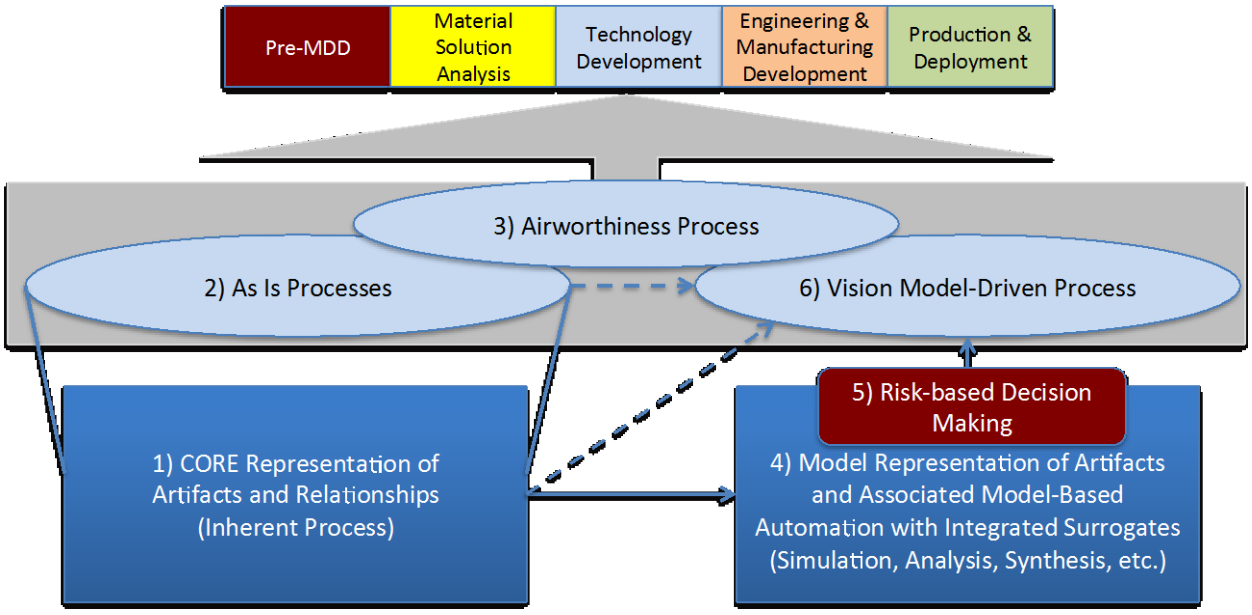


Figure 26. Model Vision at Program of Record Scope and Integrate Risk-based Decision Framework

VISION PERSPECTIVES

With a number of discussions (29 as of December 2014) behind us, it is a fairly consistent message that many organizations have not defined a Vision model. Instead they are involved in an evolutionary process of model adoption, and many want to better understand the return on investment (ROI). Some organizations do have to address some airworthiness and safety-related requirements and those efforts can lead to longer delivery schedules. Even the automakers are expending more resources in their need to address safety constraint. In addition, some of these organizations are working on a subset of the problem (e.g., V&V) [51], while others are approaching this from the contractor point-of-view, which is significantly different from that of NAVAIR. NAVAIR is working in the early stages of DoD 5000.02 [36] lifecycle (i.e., Milestone A, B, C), and they ultimately produce requirements and design constraints that are provided to the contractors. There efforts are focused on problem understanding in the context of mission analysis.

The objective for the Vision should address the questions:

Can we create models to cover every type of artifact that is required to produce a system and comply with DoD and NAVAIR processes and requirements (e.g., Airworthiness)?

Can we use model-based simulation, analysis, synthesis and generation to rapidly traverse the "Virtual Vee" and continuously, both horizontally and vertically, leverage evolving digital representations (e.g., models, surrogates) to assess the system design at various levels of fidelity in the context of continuously evolving mission scenarios?

- Notionally rendered in Figure 4 and Figure 5

How does the risk framework fit into the model?

We initially developed (as a straw man, see Section 5.7) an example model in System Modeling Language (SysML) that represented the Integrated Warfighter Capability (IWC). The example provided a common understanding that the goal of the modeled Vision is going to formally characterize all of the "data," relationships, automation throughout the entire lifecycle, including for example the relationship to data used by, and produced by modeling and simulation, analyses and other resources, as well as evidence captured within the models to support risk assessment and management (see Section 6).[16]

We used SysML, because we saw examples from NASA/JPL [5], who is the only organization that we met that has started this type of Vision model concept. SysML works for JPL, because their entire team is deeply versed in SysML. However, we are not sure about the approach for explaining our perspective as we also know that there may be many people in the NAVAIR that are not familiar with SysML. Therefore, we are using another approach that might be more "user friendly." There is a storyboard that was created with about 10 different views. We include an integrated overarching perspective that is shown in Figure 27. This image includes information containment and operational perspectives. Notionally starting top down and going clock-wise:

1. This is a **Collaborative Environment**

We envision access to this information to be done from at least three forms:

- Model editor form (raw for the expert modeler, and this could include many types of models, DoDAF, Simulink, SysML, Domain Specific Modeling, Cost model, Computational Fluid Dynamics, Risk, etc.)
- Web-based form; view of information synchronized from the "system model;" we have heard many discussion by tool companies, and a similar story about open MBEE from NASA/JPL [35], and this is consistent with technologies discussed by the commercial organizations
- This would allow for a "dashboard" type web interface, like the SETR Manager that would provide personalized live updates to the user; including prioritizing a user's workload by allowing them to see how their task affects the bigger program

Documents can be automatically generated through personalized or program-standardized templates

- Access to information is available to all team members and they can see the same instance of information as other team members so this collaborative environment, which provides a single source of truth; security mechanism and role-based view mechanisms also exist today

These types of efforts are under way at NAVAIR and more broadly throughout the Navy and other services [84]

[16] There are number of useful representations and documentation that are not currently released for public viewing.

2. There is a **Continuous Digital Thread (orange dashed line)** running through all aspects of the concept that is addressing an ever evolving set of needs generically referred to as "**Capability Sets**"

Continuous digital thread means that all digital data can be connected and every piece of digital content is aware of other digital content; this is essential for single source of truth

The modification of any item can trigger events related to all other dependencies and can change the state of that data, and related data (e.g., trigger weight analysis for entire aircraft if the wing weight increases)

3. **Containing System**, as described in 2.3.1, must represent the SoS, including environment and resources with sufficient fidelity and semantic precision to understand how a target system interacts within its environment.

Capability Sets are conceptually produced in the context of the containing system through mission-level modeling and simulation analyses to address evolving threats/needs as input from the efforts of the modeling and simulation group

We were provided details by two NAVAIR groups involved in mission-level analyses, but will not include that information in this report as it not publically released

- This is related to discussions at the Mission Level as reflected conceptually in Figure 5 (e.g., operational, and kill chain scenarios, etc.)

4. **Program of Interest** should be an ever evolving instantiation starting from elements in the Reference Model (or Reference Architecture), which are parts of the Designing System

We believe that a model-centric approach to a radical transformation will involve the use of "Model Measure" or Model Maturity Levels to assess the state of the models' completeness, well-formedness, consistency, etc. and its ability to produce all of the needed evidence associated with the Airworthiness constraints

During the iterations the capability sets should start converging to a mutually acceptable program of interest

New technologies and knowledge captured in the creation of any new system should be captured in the Designing System, including a continual evolution of reference architectures (template of knowledge encapsulation about air vehicle systems and weapons)

5. **Designing System** includes all information it takes to go through analyses and design development; this would include:
   - SETR Manager
   - Every modeling and simulation capabilities, 1D, 2D, 3D, SW, HW, System, Mission, etc.
   - Trade space analyses
   - Reference model (reference architecture) that characterize the architectural structures of air vehicle systems

Attributes associated with data about those system/subsystem/components

Airworthiness constraints

Tools that are used to provide analyses for those different subsystems (e.g., Simulink for control laws)

- Cost models linked to the reference architectural elements
- Tools such as Dakota for Quantification of Margins under Uncertainty (QMU)
- Other risk modeling
- Cost and schedule modeling and tracking
- System Integration Labs
- Surrogates, hardware, software
- New tools
- New approach for characterizing modeling maturity measures

This list and story is not exhaustive.

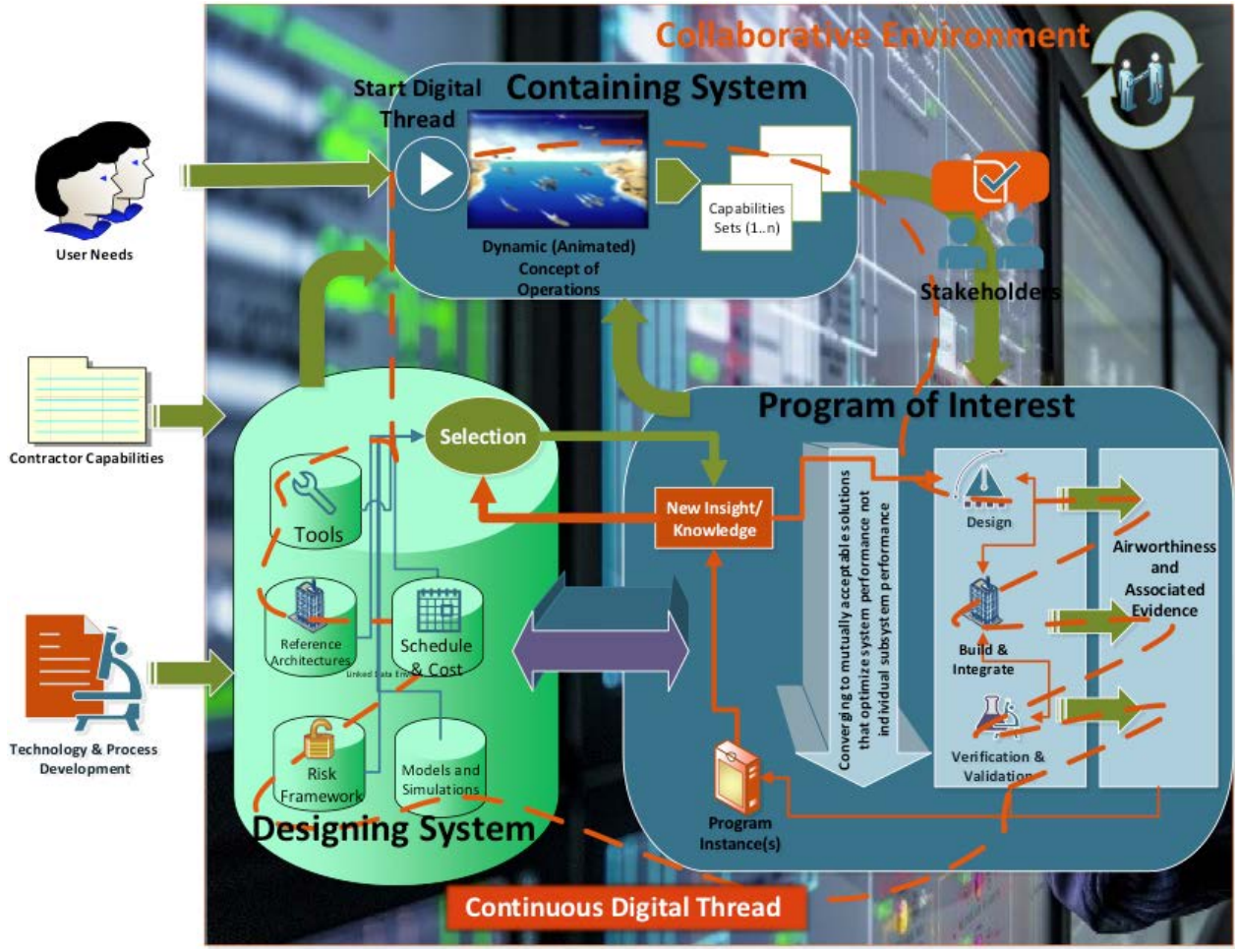


Figure 27. Overarching Concept for Vision

SCOPE TO PROGRAM OF RECORD THROUGH DIGITAL CRITICAL DESIGN REVIEW

Figure 5 puts the scope of the POR into context, as well as making the context of a POR part of an evolving platform. This too abstractly reflects on the boundaries between the POR and the mission level. The scope of this research task has been reduced to focus on the lifecycle phases up to critical design review (CDR), for the "As Is," Airworthiness, Vision model and risk framework for a POR. It was thought that the technical reviews are good "checkpoints" since they focus on different decisions and levels of engineering content that would need to be represented in the models. Only the PDR and CDR are always required. Other reviews such as: ASR, SRR, SFR, TRR, SVR, PRR may or may not be required on a given program. Ideally, we are looking for a new concept: Digital design from CDR artifacts (DCDR). We want to investigate a more continuous notion of PDR and CDR (or DCRD) where reviews of models and analysis artifacts occur "everyday" until all of the required evidence is provided in order to support contracts and signoffs; any meeting can be virtual and in real-time when data is available. This concept is now part of the new SETR Manager, which is briefly discuss in Section 5.3.2.

More importantly, now that we evidence about some aspects of the technical feasibility question, we want to understand if there are alternative types of model measure that can be used to supplement or eliminate these traditional document-centric reviews as part of the radical transformation. Part of the ongoing research is to investigate if such a concept is viable.

CONTEXT FOR PROGRAM OF RECORD SYSTEM

The context for the POR starts from environmental aspects at the mission-level as discussed in Section 2.3. For many efforts organizations often start with a DoDAF operational view (OV-1) diagram of the mission-level with systems-of-system (SoS) level interactions; increasingly many are using dynamic OV1 such as those reflected in Figure 8, which aligns better with the model-centric engineering concept. The operational views decompose the mission within the context of the situation, and provide different viewpoints that describe the tasks and activities operational elements, and resource flow exchanges required to conduct operations related to scenarios, as reflected in Figure 6.

NAVAIR Mission Level Modeling and Simulation (M&S)

We had a discussion with two NAVAIR M&S groups who are responsible for analyzing the mission scenarios. They do have a vision for the future; they indicated that there will be much more cross-domain integration, but the current capabilities appear not to have much integration. The views from these M&S capabilities (i.e. capability sets in Figure 27) define what we discuss as the **containing system** part of the Vision model, but currently they are not integrated. For our research task scoped at the POR, this information is on the interface boundary, but there is not much that feeds down today; that is, the majority of the analyses from the M&S groups are focused upwards towards the campaign level, rather than downwards towards the system (aka engagement level).

Model-centric perspectives at the POR level would be potentially useful for this effort, because their M&S capabilities must often create some type of abstraction of the PORs and platforms.

This is a plan for 2015, which is to better understand the interface boundary between the M&S level and the POR level within the context of the Vision.

NAVAIR Study Views

Study views were created to address a number of challenges at this level and in creating DoDAF requirements, discussed more in Section 5.3.1.4. The study view concept builds on lessons learned from creating early DoDAF models; analyses have uncovered that interoperating at the lowest (data) levels is insufficient for scenarios, and scenarios require behaviors; missing at the data level. DoDAF does not accommodate other scenario requirements (e.g., conditions assumptions) very well, and is insufficient to fully characterize the dynamic needed for analysis.

A mission-level SoS analysis begins with formalization through Study Views, as reflected in Figure 28, which has M&S dynamic views and visualization. Study views provide structure and a common context that acts as a basis for framing and bounding the functional decomposition of DoDAF products. Study views formalize the need and intent, provide a situational context and influencing factors to frame and bound the functions and activities of the mission and scenarios that ultimately lead into corresponding representations of the Mission and System Capabilities (i.e., the capabilities for the POR). These capability representations are further analyzed using modeling and simulation and corresponding analysis capabilities. The outputs of which are then formalized in terms of DoDAF artifacts that are formalize by the NAVAIR Architecture group, discussed in Section 5.3.1.4. This information will form the analysis boundaries for the System Capabilities information needed as requirements for the POR.

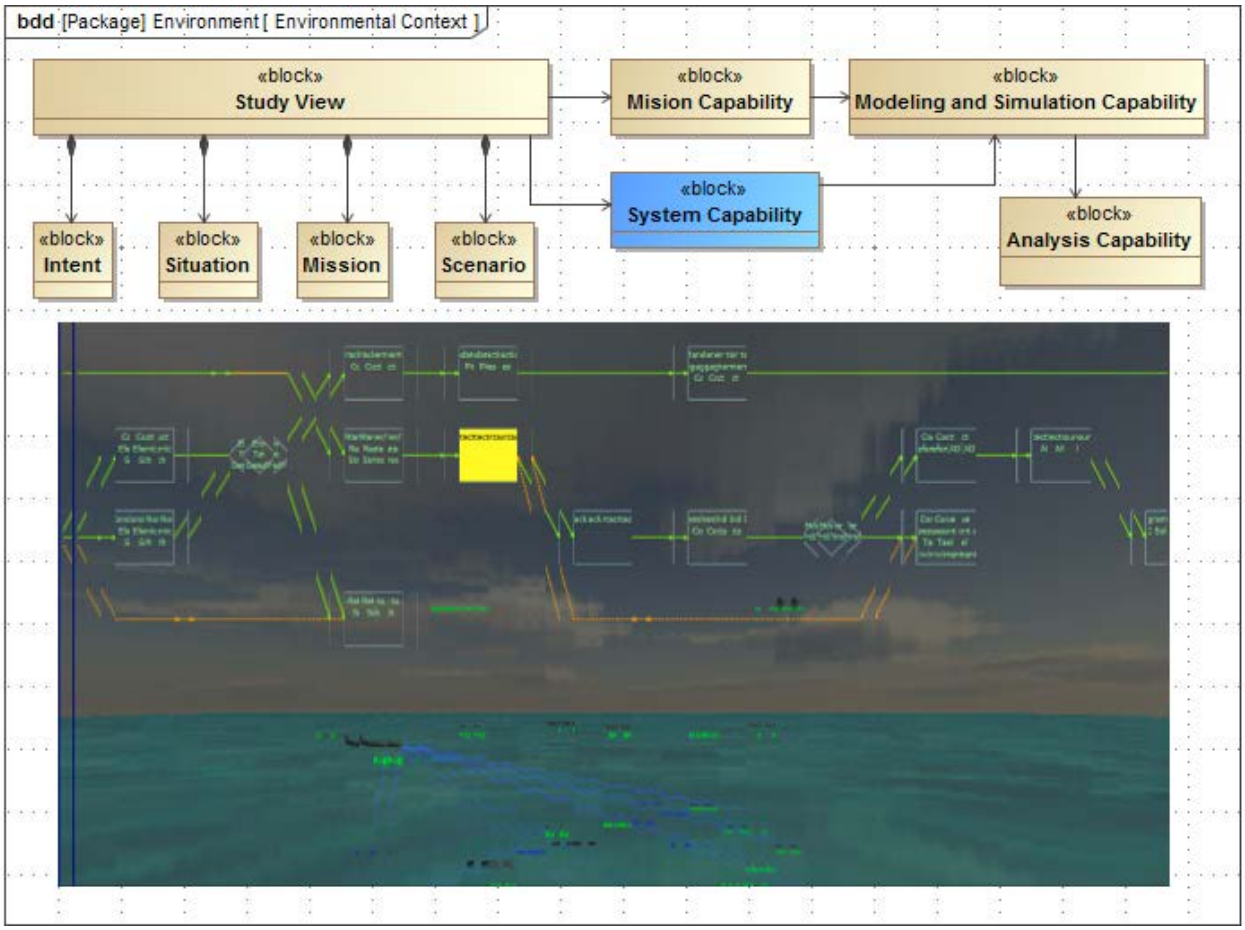


Figure 28. Mission Context for System Capability[17]

We heard a similar story that is being applied and evolved on Jupiter Europa Orbiter (JEO) project [75], and we summarize some aspects of it here, because it goes beyond what we currently know about Study Views. Like NAVAIR, they too have created their own supporting tool, discussed in Section 5.3.1.3 that provides for the structured entry and retrieval of architecture artifacts based on an emerging architecture metamodel.

The architecting focus was elevated to a more prominent and formal role on the JEO project than has been done on most other NASA/JPL projects; the emphasis is to make systems engineering's basic processes, such as: requirements generation, trade studies, risk management, design and interface control, verification and validation, etc., more coherent. The new architecting process used on the JEO project and framework is intended to aid systems engineering in the following ways:

Adding guiding structure

Providing better integration of the resulting artifacts

Ensuring comprehensive attention to important relationships

[17] Image source: Thomas Thompson, Enabling Architecture Interoperability Initiative, B210-001D-0051 Unclassified.

Facilitating broad understanding of the architecture

Maintaining system integrity over the course of development

Helping to ensure comprehensive verification and validation (V&V)

NASA/JPL acknowledged the choice of a different framework (e.g., not DoDAF, which is used by NAVAIR), because they viewed the choice of framework should be dependent on the nature of the system and circumstances it was designed to support. The JEO most closely aligns with the emerging ANSI/IEEE 1471-2000 standard [50] for software-intensive systems. The architecture artifacts include, but are not limited to, Stakeholders, Concerns, Viewpoints, Views, Analyses, Models, Elements, Scenarios, Properties, and Functions, which align with many of the Study View concerns.

The JEO project team efforts have focused on five objectives:

Identifying and capturing stakeholders and their concerns

Developing the content for and capturing viewpoints and views related to the concerns

Identifying and initiating trades that are needed in the near-term

Maturing the models that are needed to support those trades

Training for the growing architecting team

Their JEO project team MBSE efforts have used the System Modeling Language (SysML). They developed SE ontologies in Web Ontology Language (OWL) [89] to provide a way of defining a set of concepts and properties applicable to the domain of discourse; in this case not about the space domain, but about the SE domains for concepts such as: component, function, requirement, and work package, data properties like mass and cost, and object properties (relationships) like performs, specifies, and supplies. This provides for a controlled vocabulary and enforcing rules for well-formedness, which permits, among other things, interdisciplinary information integration, and automated analysis and product generation. Because the SE ontologies are expressed in OWL, the ontologies are amenable to formal validation (syntactic and semantic). The NASA/JPL project teams can use formal reasoning techniques and tools to ensure that the models are consistent and satisfiable, with respect to the ontologies, and constrained within the bounds of Description Logic[18], which ensures that certain reasoning operations remain tractable. Once a model is completed other transformations are performed to the model that can check properties such as well-formedness and consistency of the model. The NASA/JPL projects using this approach currently have about 60,000 test cases; this concept could be part of an approach to model measures in a NAVAIR radical transformation.

Reference Architecture & Model Based Engineering Environment

The NASA/JPL projects have a related reference architecture and associated open Model Based Engineering Environment (Open-MBEE) [35] that they are using and evolving on the JEO project. The reference architecture aligns with the vision model concept. They used MagicDraw, which

[18] Description logic (DL) is a family of formal knowledge representation languages. It is more expressive than propositional logic but has more efficient decision problems than first-order predicate logic.

supports SysML/UML [70] and other modeling capabilities to define activity that are transformed to an Oracle database to manage workflow. MagicDraw also provides support to plug-in domain specific modeling tools [60]. They are modeling their artifacts and activities to generate the controls for a workflow engine.

Figure 29 provides an overarching perspective on one of the views extracted from a report [6] that is applicable to the Vision model:

Blocks in the diagram define categories of items requiring exposition in the architecture description

Accompanying each category is a template (not shown) specifying the sorts of information required for each member of that category

Stakeholders and their Concerns are the drivers for everything else in the architecture, i.e., they can be considered the 'entrance points' to explore the framework

This is somewhat analogous to the purpose of the Study Views developed at NAVAIR, although NAVAIR does not have a similar representation of its context in a model representation

The Element is a place holder for aspects of the System to be designed (i.e., Program of Interest in Figure 27)

The Models, the Analyses performed on them, and the Scenarios, which relate to the "Containing System" (e.g., for a Program of Record) complete the blocks of the Architecture Description

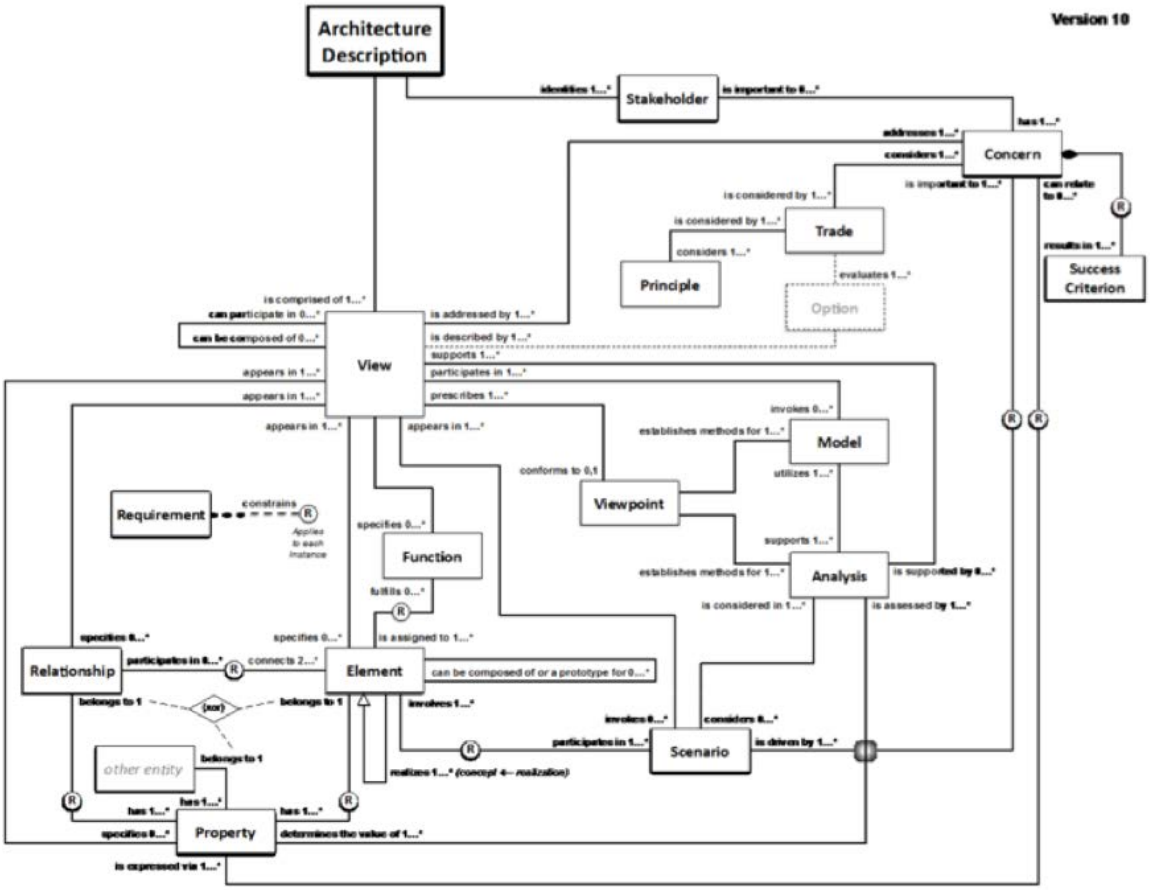


Figure 29. NASA/JPL Architecture Framework Tool (AFT) for Architecture Description

In addition, with the information provided on the reference architecture and the associated modeling patterns, this concept provides the best story we have heard as it relates to formalizing the concept of the Vision model. This perspective informed our development of a NAVAIR-oriented concept reflected in Figure 27.

NAVAIR Architecture Group

The inputs from the M&S group, such as Study Views are inputs to the System Requirements Analysis and Architecture, which focuses on developing DoDAF views to drive the system analysis and design. They are working toward the requirements for the Naval Enterprise Architecture Repository (NEAR), which includes the need for Physical Exchange Specification (PES) compliance, however this is a challenge, because some of the tools do not support PES in the same way. While these efforts are using models, they are not using dynamic models. Most important is that these DoDAF type models are increasingly being used in communications with system contractors. While this is not necessarily a radical transformation, it continues to support the concept that sharing information through models is happening today.

SYSTEMS ENGINEERING TECHNICAL REVIEW (SETR) MANAGER

This section briefly discusses the new SETR Manager, which is inherently part of the "As Is" process, but could be part of the "to be" Vision. The SETR Manager is a server/web-enabled way to navigate through the SETR checklists. It provides real-time status updates and reviews, and allows for discussion tracking providing a familiar Facebook and Twitter style that should provide an easy-to-use look and feel, allowing teams to come up to speed quickly. This capability is a transformation of a few different types of SETR checklist approaches that have structured and layered different types of tooling for the checklist with some reorganization of the checklist questions (more 5000), but layering them. The Tier 4 questions (~1500) are still Yes/No, and the other possible question have now been moved to Tier 5, and are referred to as Considerations, which add context to the Tier 4 questions. There may be a need to move some of the Tier 5 questions to Tier 4.

This will be an ongoing evolution, which they want to do in a much more iterative (Agile-way). In its current state the SETR Manager:

Provides dashboard views of the SETR Manager data for all primary management roles, and competency (tech authority, SETR content owner)

Uses the dashboard to support drill-down of data

Visualizes historical trends where possible

Allows comparisons between different sets of data (i.e. between multiple competencies or programs)

Steers attention quickly toward potential issues and/or tasks that must be accomplished

The SETR Manager is part of the Designing System, shown in Figure 27. The overall metaphor provided by the capability aligns with a much more collaborative way of supporting real-time reviews and consolidated measurements in consistent colorized dashboards, with visualization. The server-based approach allows for an easier and more continuous updates as NAVAIR adapts, and to support integration of other web-enable and server-based approaches for continuous and collaborative engineering.

While this too is not necessarily radically transformative, we believe that this type of interface is a complement to native modeling environment for a Vision model. It plays an important role in presenting management information today, but as a server/web-enable mechanism provides a surrogate for presenting document-like information can be directly generated from underlying models.

"AS IS" PROCESS

The NPS team is modeling the "As Is" process. The model includes a large number of the artifacts that are produced as part of the SETR process. A key guideline for the SE process is the SETR process as it characterizes many aspects of the information that needs to be collected through questions and associated checklists. The team also examined most of the required artifacts that

are produced in their current processes. Our NAVAIR team categorized about 330 artifacts and is now realigning its modeling effort. This is currently developed in a CORE model.

The artifact analysis resulted in a somewhat abstract understanding of the "As Is" process. Many of the artifacts are just named items with no formalized definition of the artifacts or resources used to produce them. The second phase of the effort has been trying to extract knowledge from stakeholders that have used the processes to further refine both the artifacts and overlay a process. Additional details can be found in the RT-48 technical report [15].

AIRWORTHINESS FORMALIZATION

The Airworthiness process is used to ensure that the necessary evidence is provided in order to get a flight clearance. Brian Nolan from Solute is working with Richard Yates from MITRE to create a model for this process. They have used authoritative sources [61] [64], however, a significant amount of guidance is obtained through interactions with Airworthiness subject matter experts (SME). They are using those discussions to model the Airworthiness aspects and related it to the "As Is" process. Some of the reasons for modeling this process:

Currently is document based; mostly manual retrieval of required data/evidence

Sensitive to "personalities" of performance monitors

Heavily dependent on implicit expertise and "tribal" knowledge

Inadequate guidance on types of verification required/feasible

Mass of work required makes it expensive and time consuming—how to reduce time and cost, or at least be more sensible about cost

The Engineering Data Requirements Agreement Plan (E/DRAP) is another essential artifact that is used in flight readiness assessment. A possible approach is to decompose E/DRAP as a metamodel (all of possible artifact classes and their relationships). Normally the E/DRAP is done in terms of allocated baselines that characterize both the operational effectiveness and operational maturity. We recommended that we work backwards from all E/DRAP-required information classes that are needed for airworthiness decisions, and them characterize the inputs and associated processes required to produce the E/DRAP information. Remembering that the Vision model is about system data and information that must be produced to go through the different decision gates, therefore, we need to hear more about the E/DRAP to better understand how it would relate to a Vision model.

The following provides some highlights on the current approach, which is being worked in 2015:

It is unclear if Airworthiness is actually a process or a set of constraints

- If there is a process, it could be characterized in a general way and could be applied to every decision, for example:
  - There is a Task to make a decision
  - A Task is performed by some Actor that has some type of Role
  - A Task takes Artifacts as Inputs

  - A Task produces Artifact as Outputs about evidence to support a decision
  - Artifacts are linked to various types of sub-artifacts/information/data
- A generalization of the above was represented in a class diagram related to the information provided above (a type of model architecture as it is a general repeatable template that could be applied to any decision)

Represent needed evidence using a small set of views in SysML diagrams[19]

- A SysML Block Definition Diagram (BDD) decomposes the structure of all of the artifacts related to evidence and associated with authoritative roles
- As is common in a reference architecture that BDD structure maps to an aircraft and the decomposed subsystems/components, each with details defined as attributes such as the weight
- Constraints are associated with those attributes (e.g., the weight of the wings <= [TBD value for specific aircraft], the overall weight of the aircraft <= [TBD value for specific aircraft])
- Evidence to support the airworthiness, would show that the constraints are met, potentially with certain amount of margin
- We discussed that this should map to the Engineering Data Requirements Agreement Plan (E/DRAP)
- The constraints could be formalized in SysML parametric diagrams
- Parametric diagrams define constraints related to attributes for blocks in BDDs

Our sponsor wants to frame this modeling effort as part of the risk assessment approach to the Airworthiness stakeholders. In Section 6 we provide an example of how to frame Airworthiness in the context of the Military Standard 516C [61] as a Bayesian model (see Figure 37) for risk assessment.

The formalization of the Airworthiness information is not only needed for the “As Is” process, but it generally applies to the transformed process too as reflected in Figure 27. The proposed approach is being characterized using formalizations that can be linked via web-based interfaces to people having different roles in the Airworthiness process.

If we envision in the future state that NAVAIR will capture reference architecture representations of air vehicle systems, then the constraints characterized in this approach for Airworthiness can be directly associated with the attributes of the aircraft system at various levels of the system architecture. The specific instances of any system would make those particular constraints relevant to the airworthiness process, and based on our understanding would also be associated with the E/DRAP. The relationship to the reference architecture also helps:

Provide a perspective on the coverage for an entire air vehicle system

Allocate to the different subsystems and competencies and map to the particular roles of the various SMEs

[19] These diagrams are in a documented with a Proprietary Notice and therefore are not include here.

Shows dependencies and relationships between different subsystems (in architectural views)

- "Glues" all decision together at many levels

Reference architecture is also related to the questions now being captured in the SETR Manager

We have continually talked about the reference architecture as part of the Vision, and at this point it is fully acknowledged by all that it is necessary, but "How" is the question

---

MODELING AND TOOLS FOR THE VISION

Our team has had numerous discussions about modeling representations, languages and tools for the Vision. The examples in Section 5.7 use SysML, which is a standard modeling language. It is general [51], but there are limitations. The basic SysML diagrams in the modeling environments are mostly static. System engineering models defined in SysML are descriptive in nature and do not directly produce analytical results [51], nor are they executable [25]. Different tool vendors provide extensions or their own analytical capabilities that solve SysML parametric diagram [19]. Since the parametric relationships are solved as a system of equations, the analytical model is limited to simple equations. To be able to use more sophisticated engineering analyses, external analysis tools need to be connected to SysML models. Other research efforts are attempting to leverage other standard modeling languages such as Modelica [70] that have a broad range of analytical support through integration between SysML and the Modelica. Modelica is a standardized general-purpose systems modeling language for analyzing the continuous and discrete time dynamics of complex systems in terms of differential algebraic equations. Domain Specific Modeling environments (e.g., Simulink for control systems) often have richer semantics (e.g., structure, behavioral and sometimes temporal) to support dynamic analyses and simulation; some also have formal method analysis and automated test generation [9] [17] [74]. Other approaches provide process integration and design optimization framework allowing for many available analysis solvers or custom solvers for all type of analysis with simulation and workflow automation [55].

There are many modeling language and tool options available to us. This overview is not an exhaustive list and the specific modeling language and tool(s) for the Vision model has not yet been decided. Because SysML is general, there are possible mappings to many types of modeling languages (as is true for UML also) [92] as well as support for programmatic interchange based on the XML Metadata Interchange (XMI) standard [69]. This may rationalize why some organizations are using SysML as an integrating framework, that is, they may not be modeling in SysML, but they are using SysML (and associated tooling) as a mapping or an interchange medium between different modeling languages and environments [5] [35]. While the SysML and UML languages and tools help significantly to formalize the expression, exchange, and graphical representation of system models, SysML and UML languages remain ambiguous and in need of extensions to capture the specific semantics of a given engineering domain [84].

Even with the concerns about the understanding of SysML, discussed in Section 2, our team will use modeling notations like SysML in this section of the report. However, the perspectives cited

in this section reflect on why the Vision must go beyond and use other more semantically rich and precise model representations, as well as supporting semantically consistent model (digital) interchange between different simulation and analysis tools. Our efforts planned for Phase II will investigate a potentially more general approach for representing the Vision, which we think can support the entire lifecycle [88].

STRAW MAN

During the kickoff meeting, it was decided that we would attempt to build a model of the Vision. Therefore following good modeling guidelines, we started with a context-level representation that was derived from Integrated Warfighter Capability (IWC) graphic associated with Task 3 shown in Figure 1. The top level IWC is represented using a SysML Block Definition Diagram (BDD) diagram as shown in Figure 30, and provides a way to reflect that the effort involves characterizing all types of information that is necessary to design, verify, validate, produce, acquire and deploy a weapon system. We used other documents describing the Operational Concept Document of Navy Integration and Interoperability (I&I) Environment [63] and created a similar diagram as shown in Figure 31. Regardless of the content and modeling approach (SysML), the mere existence of these examples stimulated significant discussion at the working session and clarified for the team what is meant by modeling the Vision and the concept of capturing all information in "system" model.

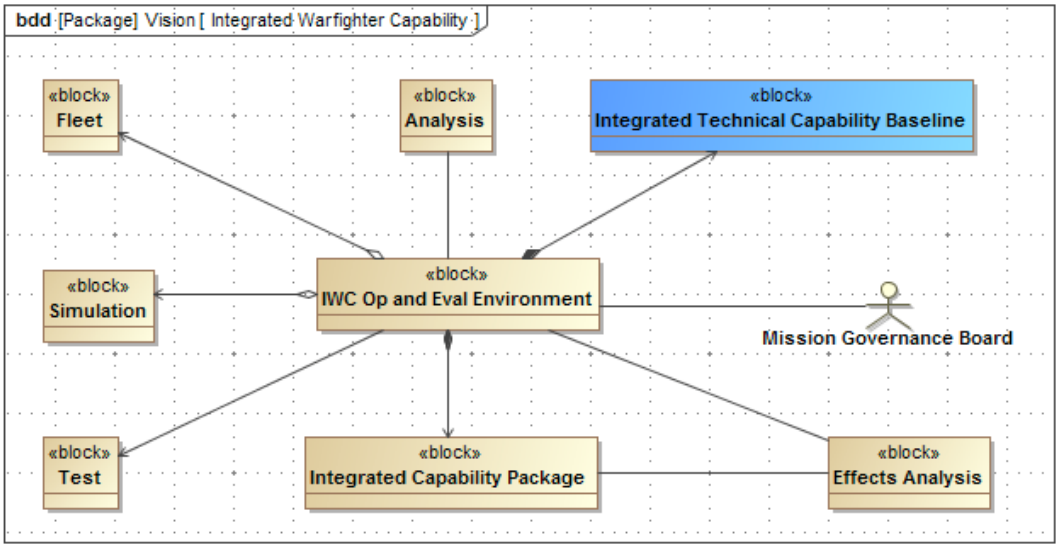


Figure 30. SysML Context of IWC Vision

As reflected in Figure 27, the Vision model will be a reference model (aka reference architecture or metamodel) of a multi-level, multi-domain integrated engineering approach to support the IWC. It is not going to describe a specific instance of a system; instead it will ideally characterize all of the types of information related to the design including characterizations of the supporting environmental resources for simulation and analyses, design parameters and constraints, verification and flight readiness evidence, and associated risk-based signoffs. Ultimately, it should

include everything to the Bill of Material (BOM) required to manufacture and produce the system (or in the future the specifications for 3D printing).

It was decided to scope the effort at a Program of Record (POR) (e.g., F18 with weapons, CH-53). Referring to the BDD in Figure 30 and Figure 31, a POR relates to the Integrated Capability Technical Baseline (ICTB) block. The ICTB block is also represented in Figure 30 (the Integrated Warfighter Capability BDD). From the perspective of the Integration and Interoperability (I&I) Environment BDD, relationships from the ICTB block to the Mission Technical Baseline (essentially where the requirements for the ICTB are derived), and System/Program Technical Baseline blocks are reflected. All of these blocks relate to the I&I Repository. The I&I Repository is part of the Navy's Integration and Interoperability Integrated Capability Framework vision that includes an enterprise data environment for storing and sharing DoDAF architecture and other supporting I&I data in a common format with common ontologies to support cross-correlation and alignment [63]. These BDDs provide two perspectives on the relationships to the ICTB within the NAVAIR vision, but this is still at a very high level. In order to complete a representation of the Vision it will be necessary to formalize:

All information as typed data elements, which can be represented as attributes in a model

Data flows reflecting the data dependencies between blocks

- BDD diagrams often have associated Internal Block Diagrams (IBD), which show hierarchically lower-level diagrams with the corresponding data flow between the lower-level blocks
- As another type of example, Figure 32 shows that the Vision must not only be able to characterize the elements of the vehicle system, but should also characterize the elements within the overarching environment that show uses or dependencies to resources such as simulation, test environment, instrumentation and logging
- Surrogates would also be represented by blocks

Control flow reflecting both sequential and concurrent flows

- Activity diagrams in SysML can represent both control flow, and the associated data flows that would be associated with flows within an IBD

There are other behavioral views (e.g., sequence and state diagrams) and constraint views (parametrics) that would be necessary to fully characterize the information needed to produce an air vehicle system.

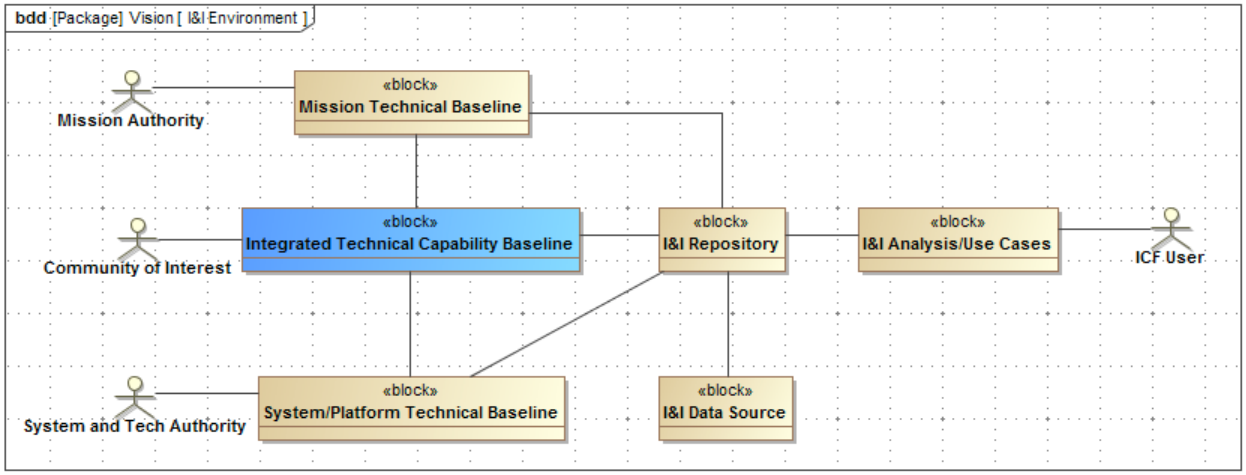


Figure 31. I & I Environment

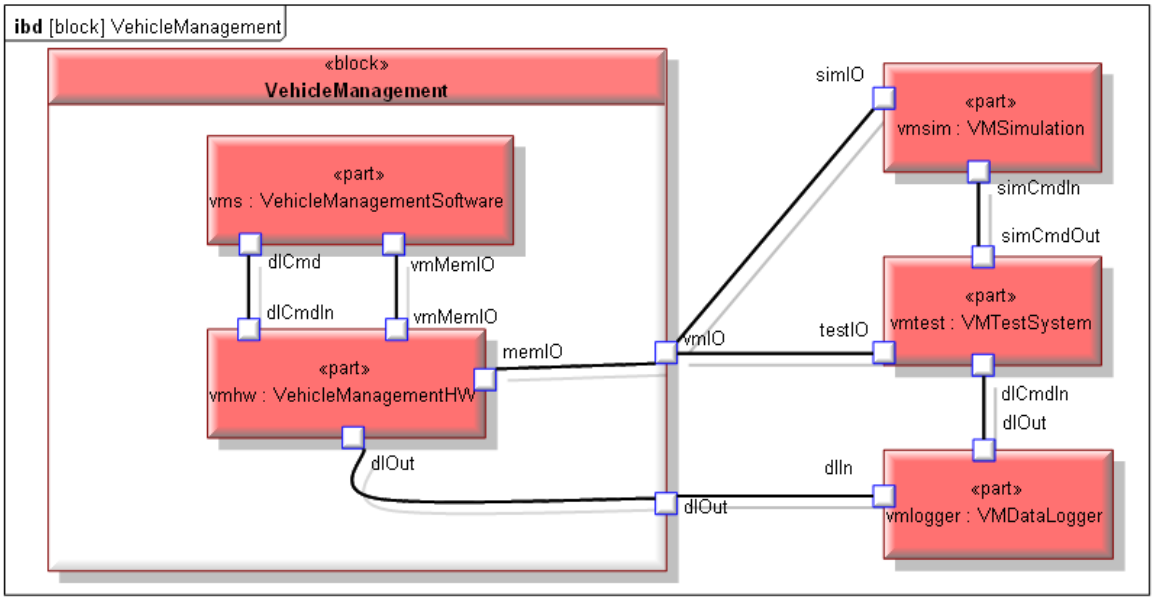


Figure 32. Vision Model Interface to Simulation and Test Resources

MODEL-CENTRIC ENGINEERING PERSPECTIVES

Section 2 provides some perspectives on model-centric engineering. This section provides additional information related to discussions or actions from our working sessions that relate to what it means to do model-centric engineering.

MODEL TRANSFORMATION RATHER THAN MODEL EVOLUTION

To reflect on the concept of model transformation rather than model evolution, we provide the following example to describe how model-based automation can completely eliminate manual effort and result in radical transformation of the "As Is" process through an automated workflow. The following provides a scenario for how to think about using models to replace artifacts, and

more importantly how model-based automation subsumes manual efforts [13]. This process was used in the verification of the control laws for the F-35 [67]. This scenario relates to an "As Is" artifact called the "Flight Control Detailed Design Report." In a model-centric world this type of artifact would:

Represent "Control Law" in a model

- Simulink[20] and Stateflow are commonly used to model control laws (e.g., F-16, F-18, F-35)

Automated analysis that exists today, (e.g., it has been applied to F-35) would include:

Satisfiability: proving that each thread through the model has no contradictions (mathematical consistency)

Simulation

- Simulation of Simulink models is often done using Matlab
- Support high-fidelity simulation using Matlab
- Support higher fidelity with real-time execution within the surrogate or prototype system implementation or actual hardware though automatic code generation

Synthesis or generation

- Code generation from Simulink models can be provided by Mathworks and other commercial products
- Automatic test generation directly from Simulink models
- Automatic test driver generation

The test vectors are transformed into various types of test drivers that are run both against a Matlab simulation and the auto-generated code; if all tests pass (the actual output equals the expected output) in both the simulation and generated code execution environments then there is a strong verification argument that the code satisfies the specification

- Organizations run the test through both the simulation and code, because organizations have been able to find errors in the code generation (Risk reduction argument for using model-based tools)

Code coverage tools such as LDRA and VectorCAST have been used to show that the tests provide Modified Condition/Decision (MC/DC) coverage

- Code coverage measurement, which provides quantified risk reduction evidence

The Mathworks code generation uses a particular algorithm that produces code that is "deadlock" free

- Eliminates concurrency analysis

[20] We are not promoting Simulink, we use it as an example, because it is almost a defacto standard for control system modeling and simulation, and it was the tool used in the above scenario.

These are types of model-based automation that leverage models to “Cross the Virtual V.” While this can be and is commonly done on low-level high-fidelity models, we are also interested in applying this type of concept at the upper-levels of the “V” with varying levels of fidelity that provide integration of model and model automation at different levels of the “V.”

This is a positive story as it relates to the use Simulink-based modeling tool chains that can significantly reduce time by both supporting simulation, code generation, analysis and test generation. However, other forms of software modeling have not had this same type of automation, because behavioral information in a modeling framework (e.g., UML Rhapsody) is manually coded, and that cannot be analyzed in the same way that Simulink models. This is a concern as software is growing in complexity and size. This is related to the challenge areas discussed in Section 2.5.1.

CROSSING THE VIRTUAL “V” BY LEVERAGING MODELS, DIGITAL AND PHYSICAL SURROGATES

We have continually discussed the notion of “Crossing the Virtual V” as an important way to assess system design concepts in the context of mission scenarios. However, in discussions with organizations, there are some that believe that the notion of a “V” is a historic manifestation of “the” traditional flow of document-driven work, and we should eliminate the use of the ”V” as part of systems engineering dogma as it is counterproductive to embracing approaches that support the continuous integration of digital artifacts. The “V” introduces points for disconnects and failure. What is more critical in the Vision is continuous integration of various types of digital assets with varying levels of abstraction and various degrees of fidelity as reflected in Figure 4. Section 2.3 provides some discussion on this point using examples to further clarify the notion of physical surrogates, and support the argument that the “V” may not be a good metaphor.

The concept of model-centric engineering relies heavily on digital assets such as physical surrogates, existing system, or component re-purposed for new concept exploration and prototyping. Our NAVAIR team created a concept for representing System Level Maturity. It reflects on the idea that as we are attempting to “Cross the Virtual V” and will rely on physical surrogates, which is commonly done today, both in aerospace and other domains, such as auto racing. The actual airframe, shown Figure 4 along the bottom matures (right-to-left) and the actual aircraft is first flow (e.g., F-35, 15-December-2006) long before many of the complex software intensive systems are developed and integrated, as the aircraft airframe and new materials are being evaluated. Key early capabilities such as software for the control laws to fly the aircraft are often evolved from earlier aircraft systems (e.g., many versions of MATRIXx and/or Simulink models have been evolved for years, and will continue to be evolved for years). Yet, all of these systems are continually refined and as the timeline of system capabilities mature, new capabilities are added to the system. We believe that in model-centric engineering, it will be possible to have continuous integration and tests, much like agile is used in software. Formalized interfaces are required for integration, and the semantics for the interfaces often need to be formalized: 1) structurally, 2) behaviorally, and 3) temporally, in order to use surrogates and simulations. Document-based specifications do not formalize these, however some modeling approaches can, and with semantic formalization, automated verification can be supported directly from the models.

REFERENCE MODEL

We have heard from many different organizations about the use of platform-based designs (see Figure 9) and reference models (aka reference architecture). As shown in Figure 33, there are many types of major systems elements and subsystems within an aircraft system. Therefore, we can image that the model of the Vision must need to use a type of reference model in the characterization of an integrated set of model types (software, electrical, hydraulic, human interface, etc.) that represent all of the engineering aspects of an aircraft system (i.e., the "total" system model). There must also be ways to characterize different types of elements, for example, a winged aircraft may not have rotors, and a UAV may not have avionics displays.

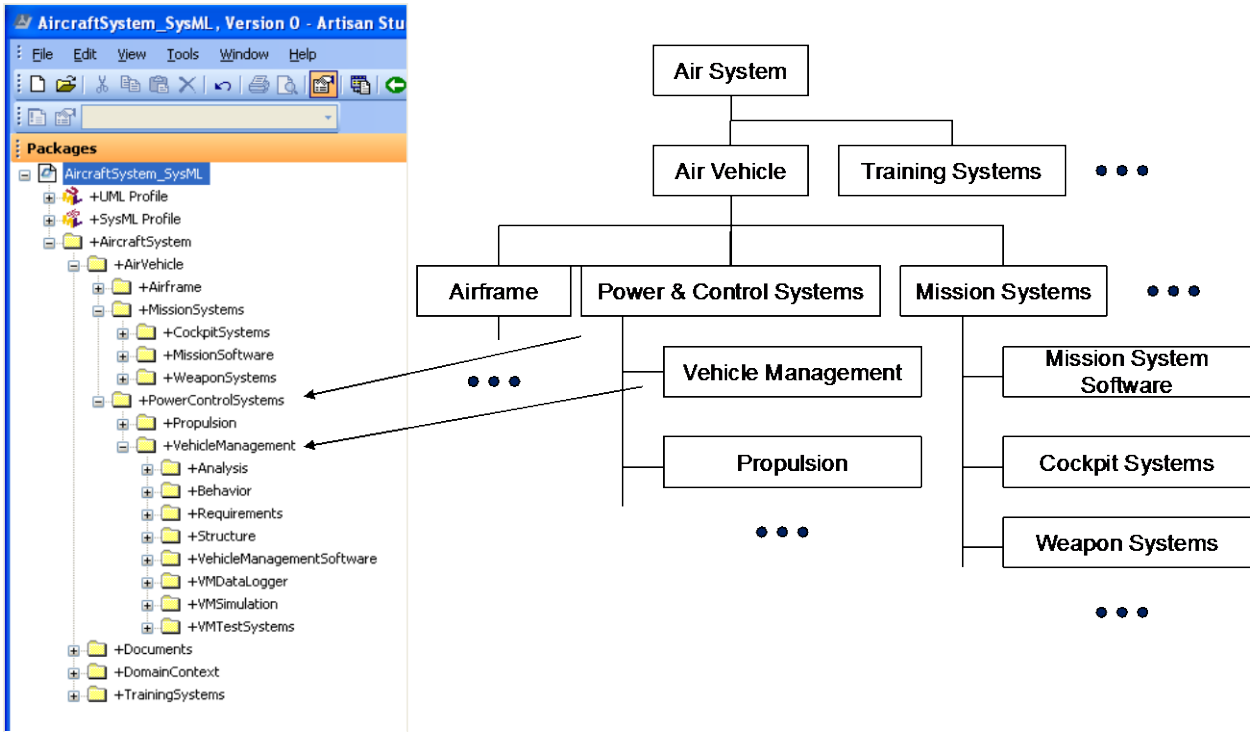


Figure 33. Model Topology Often Mirrors Architecture of System

INTEGRATED FRAMEWORK FOR RISK IDENTIFICATION AND MANAGEMENT

We are researching strategies, methods, and tools for a risk-based framework that aligns with the Vision model concept through model-centric engineering (MCE). This involves how the Vision model should include integrated risk identification and management. While there are many classes of risks to manage, for NAVAIR there are fundamentally two key classes of risk that we have been asked to consider:

Airworthiness and Safety (most critical in Technical Feasibility assessment)

Program execution (cost, schedule and performance)

There are also two complementary views of model-based acquisition with respect to risk:

Risks introduced by modeling deficiencies and risks reduced by enhanced use of modeling

Modeling to predict or assess risks (i.e., modeling for uncertainty quantification in acquisition and in the use of models)

We want also to understand how a risk framework addresses the sponsor's question:

If we are going to rely more heavily on model-centric engineering, with an increasing use of modeling and simulations, how do we know that models/simulations used to assess "performance" have the needed "**integrity**" to ensure that the performance predictions are accurate (i.e., that we can trust the models)?

This brings in the need for approaches to what has been traditionally referred to as Verification, Validation and Accreditation (VV&A) of modeling and simulation capabilities. VV&A, in principle, is a process for reducing risk; in that sense VV&A provides a way for establishing whether a particular modeling and simulation and its input data are suitable and credible for a particular use [39]. The word tool qualification and simulation qualification have also been used by organizations regarding the trust in models and simulations capabilities.

There is also a concern that the risk of SE transformation to MCE will fail to provide an efficient, effective and reliable alternative to the current process. This is an important subject, but not address in this section.

This sections:

Puts risk into context for this discussion

Discusses risk consequences from model centric engineering

The scope of the risk framework, which is fundamentally based on using model centric engineering in assessing risk

Modeling, methods and tools for quantification of margins under uncertainty

Risk-informed predictive models for risk identification based on subjective information

Model validation and simulation qualification

Risk in a radically transformed and collaborative environment

RISK CONTEXT

Defined in the DoD Risk Management Guide [38],

> Risk is a measure of future uncertainties in achieving program performance goals and objectives within defined cost, schedule and performance constraints. Risk can be associated with all aspects of a program (e.g., threat, technology maturity, supplier capability, design maturation, performance against plan). Risk addresses the potential variation in the planned approach and its expected outcome.

Risks have three components:

> A future root cause (yet to happen), which, if eliminated or corrected, would prevent a potential consequence from occurring
>
> A probability (or likelihood) assessed at the present time of that future root cause occurring
>
> The consequence (or effect) of that future occurrence

A future root cause is the most basic reason for the presence of a risk. Accordingly, risks should be tied to future root causes and their effects. A risk framework needs to address how MCE can identify future risk and characterize its margins and uncertainty in the face of continual change of the problem analysis and design.

RISK OF CONSEQUENCE FROM MODEL CENTRIC ENGINEERING TRANSFORMATION

A concern is risk of adverse consequences resulting from radical transformation to MCE acquisition. Possible adverse consequences of concern are (a) failure to produce aircraft that can be certified as safe and airworthy, (b) failure to be able to certify airworthiness and safety, and (c) certifying unsafe or unworthy systems as safe and airworthy. We are not addressing the risk that MCE transformation fails to produce the desired reduction in acquisition time and cost.

We assume that radical transformation to MCE acquisition will ***not*** involve radical change to the airworthiness certification criteria (e.g., MIL-HDBK-516B/C [61]), or system safety goals, objectives and analysis framework. However, we do believe that the production of the evidence needed will be done in a very different way derived primarily from models and the associated analytical means.

We assume that transformation to MCE will have several major effects on the airworthiness and safety certification process. We assume that manual reviews and analyses of paper-based requirements, design, engineering and manufacturing documentation will be replaced with analysis of executable models and analysis using executable/dynamic models (i.e., analysis of the models, and analysis with the models) with interactive visualizations [33]. We assume that test design and analysis, at all levels of the system, will be conducted in an iterative process in which models will be used to define the conditions for the next test (experimental design) and to analyze the test results. We assume that models of the system, models of the test process and instrumentation, and models of the uncertainty in the system models will be used to define tests that will produce the greatest possible reduction in (a) uncertainty regarding airworthiness and

safety, and (b) reduction in uncertainty with regard to the validity of the models and the inputs to the models. We assume that results of the testing will be used to refine the models and their calibration data, as well as being used to score the system with respect to airworthiness and safety certification criteria.

The risk assessment framework consists of identifying the major areas or types of risks resulting from transformation to MCE, and assessing whether those risks are manageable (i.e., the feasibility of effective risk management). Risk management consists of identifying risks, quantifying, planning and implementing detecting, mitigating, and monitoring detection & mitigation, and estimating uncertainties in them.

Some major risk types and areas that we identified are:

Models do not have adequate resolution, completeness and fidelity to be used to address the airworthiness and safety criteria

Models do not have adequate fidelity with respect to the manufacturing process, manufactured test articles and test implementation

Models of human behavior and performance are inadequate with respect the range of human errors and processing limitations, and simulators of man-in-the-loop testing fail to adequately simulate the phenomena in the operating environment

Adaptive nature of the iterative model-test-model cycle leads to homing in on specific areas of uncertainty while avoiding/ignoring others

Unstated assumptions in the airworthiness criteria and in the models are inconsistent and incompatible

Process of model calibration and validation with respect to airworthiness and safety concerns requires the same procedures, tests, reviews and analyses as the current airworthiness and safety process to achieve the same level of certainty

Model-centric airworthiness and safety certification will require more effort and a different skill set than the current process

Model-centric approach will conceal “blind spots” – factors and effects not included in the models will be ignored or concealed in certification, test design and analysis

Calibration and validation strategies for highly non-linear events and limited test & observation opportunities

Models used out of context, outside validation & calibration

Limitations, assumptions, and phenomena omitted, not often not well articulated

Deterministic chaos phenomena where small change in boundary conditions (inputs) produces rapid divergence in outputs not reflected in models or simulation scenarios

Sensitivity to complex and often unknown boundary conditions

Gaps in understanding multi-scale, multi-physics phenomena, potentially due to limitations in cross-domain model integration

Human behavior, knowledge, cognition – flight safety, damage control

Level of modeling and simulation different form level of analysis and decision

Incompatible scope, resolution, terminology with test procedures

The standard for acceptance is that the model-centric process is not worse, not less reliable, than the current process in any area or aspect of airworthiness and safety certification.

It is our opinion that these identified risks are potentially manageable. However model calibration [59], validation and accreditation for MCE with respect to airworthiness and safety may require significant effort and expertise [39]. The airworthiness certification handbook (and its expanded version), and lessons learned from previous airworthiness and safety assessments provide detailed, but incomplete, insight into the resolution and fidelity needed in the models. There has been significant progress in high-resolution man-in-the-loop simulators, airworthiness compliance verification via simulation, and formal model verification and completeness processes.

FUTURE ROOT CAUSES

As the focus of the effort is on what is problem understanding, including pre-milestone A through CDR, it will be important to understand MCE approaches to assessing the potential future root causes of risk especially as the adversaries are attempting to leverage unexpected future concerns, for example:

Adversaries adapt to avoid our systems' strengths and exploit their limitations by their choice of battlefields, tactics, and equipment

"Long-Lived" DoD Systems

Systems design to be adapted to counter adversary adaptations and exploit maturation of our emerging technologies

To deter and defeat current threats

To enable cost-effective upgrade & adaptation

This is not an exhaustive list.

SCOPE OF THE RISK FRAMEWORK

We worked with our NAVAIR team members to determine the scope for the risk framework. Key to the representation of the models (and Task 3) to support risk identification and management is to characterize the **types of evidence** that are required for **Flight clearance and Flight readiness.** It is important to understand how the models are developed and derived in order to understand the risk strategies that must be in place for **identifying** and **assessing the evidence for flight clearance**.

The process for risk under consideration for this SE transformation covers system development from Milestone A to CDR (at least for now). These questions related to risk also helped to refine

the scope for Task 3, and introduced a new term Digital CDR (DCDR), with a heavy emphasis on **digitally-derived evidence for airworthiness and safety, but to also include program execution.**

In both preliminary discussions with organizations and our NAVAIR team, it is recognized that it is important to quantify "margins" and "sensitivities" and "uncertainties" as a way to quantify risk.

> As an example, one of the organizations (in our preliminary Task 1 discussion) creates new types of advanced material for a system. They cited a particular effort working with advances in new material and processes at the nanoscale. At the component level the margins seemed acceptable. However after composing the components, margins propagated to unacceptable levels in the final integrated form.

Risk implies probabilities of what might go wrong or might not happen (on time or due to the degree expected), and some distribution of magnitude of consequences. This requires "impossible certainty" of the degree of uncertainty and advance knowledge of the likelihood and effects of unidentified events and factors. Therefore, we suggested that a better framework might be to work in terms of design margin. Design margin is more closely related to design. Design margin is how much room there is for a subsystem or component to perform less well than expected or to have greater burdens than expected until it becomes a problem. In some cases, e.g. weight, any increase adds to total weight, so instead of a weight margin, we might want to think in terms of sensitivities (sensitivity in increase in total weight, time, cost, etc. to a percentage increase in the component weight, time, power draw, etc.). This creates a number of questions for this task.

For example can we use models to see how much design margin there is in a system? Specifically when we cannot push the system to failure; which types of models and how can we use them to estimate the conditions under which the system begins to exhibit unstable response.

> In control systems analysis this is often taken to be the 3dB point – the frequency of input variation at which the output-to-input ratio is half what it was for low frequency change, or the 90-degree phase-shift point, where the frequency of input variation at which the system response lags by 90 degrees
>
> Control systems analysis methods also address the acceleration, velocity and displacement limits at which the system dynamics change
>
> Failures are often associated with transitions from linear to highly non-linear regimes; often the structure, interactions and/or dynamics change in these regions (e.g., insulators or isolators fail, etc.) – e.g., the acceleration, velocity and displacement limits at which the system transitions from linear to non-linear response
>
> Models that are relevant in the "linear" regime will give erroneous results in the non-linear regime
>
> Models that do not represent the dynamics that change the structure of a system (e.g., insulation wearing off causing a short-circuit, structural failure of a linkage, strain transitions from elastic to plastic deformation, etc.) will give erroneous results

Mechanical or electro-mechanical control and isolation systems are good examples, and important for airworthiness. Control systems work within a limited range. Standard control system analysis examines the frequency response and looks for the 3dB frequency, i.e. the frequency at which the transfer function is half of the low-frequency value (the transfer function is just the ratio of output-to-input). Other limits include maximum displacement, velocity and acceleration – when the system hits hard-stops, current limits etc.

Surrogates can be driven with increasing frequency inputs to find the 3dB point without having to experience the failure. The input parameters of virtual models are often "tuned" to match the 3dB point of test data, and then used to extrapolate to find the 3dB point of hypothetical systems. Physically realistic models can be used to estimate the limiting thresholds of stable response, provided the models and inputs are adequately calibrated and validated. Special consideration is needed for basic physical processes with non-linear response in the regime of operation, e.g., friction between moving parts versus friction between stationary parts.

Nested control loop models have been used effectively in system safety modeling and analysis [56]. The outer control loops detect changes in the response behavior of inner control loops, and then adjust the parameters of the inner control loops to bring the inner loops back into the stable regime.

In the use of modeling and simulation, there are different types of simulation with different levels of fidelity. A significant challenge is that tools do not often map well to different levels of abstractions. These are areas to frame risk. There are increasing uses of model transformation from one level or to different disciplines. Model transformation and model consistency between these views becomes a risk issue.

A companion concept is credibility of the estimates of performance, cost, etc. High credibility if it has worked in a surrogate system, less if it is similar to something demonstrated in a surrogate and model extrapolation. It will be important to better understand model extrapolations.

Less credibility the farther the model extrapolation is extended

Less credibility going from surrogate system to bench testing, etc.

Use of multi-scale calibration and validation

Use of progressive model-based design confirmation in technical reviews

- Subsystems mature and are integrated at different rates
- Sometimes early decisions are needed for long-lead time items whose specifications can be confirmed before other aspects of the system (e.g., final control system parameter values)

MODELING AND METHODS FOR UNCERTAINTY QUANTIFICATION

Sandia National Laboratory discussed some advanced approaches for supporting uncertainty quantification (UQ) to enable risk-informed decision-making. Their methods and tooling address the subjects of margins, sensitivities, and uncertainties. The information they provided reflects on the advanced nature of their efforts and continuous evolution through modeling and

simulations capabilities that operate on some of the most powerful high performance computing (HPC) resources in the world. We heard about their HPC capabilities, methodologies on Quantification of Margins under Uncertainty (QMU) and an enabling framework called Dakota, and the need and challenge of Model Validation and Simulation Qualification. They also discussed the movement towards Common Engineering Environment that makes these capabilities pervasively available to their entire engineering team (i.e., the designing system in our terminology, see Section 5.2).

We think their capabilities provide substantial evidence for the types of capabilities that should be part of the risk framework. This section provides additional details.

DAKOTA SENSITIVITY ANALYSIS AND UNCERTAINTY QUANTIFICATION (UQ)

The Dakota framework supports optimization and uncertainty analysis [80]. There is significant demand at Sandia for risk-informed decision-making using credible modeling and simulation:

Predictive simulations: verified, validated for application domain of interest

Quantified margins and uncertainties: random variability effect is understood, best estimate with uncertainty prediction for decision-making

Especially important to respond to **shift from test-based** to modeling and simulation-based design and certification

- This gets to an important point about how to use models as opposed to testing, which is critical for NAVAIR's objective to rapidly and continuously "cross the virtual V"

The HPC capabilities comes into play as they are built to take advantage of the HPC environment and can be combined with predictive computational models, enabled by environment and culture that focuses on theory and experimentation to help:

Predict, analyze scenarios, including in **untestable regimes**

Assess risk and suitability

Design through virtual prototyping

Generate or test theories

Guide physical experiments

Dakota is referred to as a framework, because it is a collection of algorithms supporting various types of integration through programmatic (scripting) interfaces; this is very representative of the concept of **model-centric engineer**, see Figure 34. It automates typical "parameter variation" studies to support various advanced methods (discussed in Section 6.3.2) and a generic interface to simulations/code, enabling QMU and design with simulations in a manner analogous to experiment-based physical design/test cycles to:

Enhances understanding of risk by quantifying margins and uncertainties

Improves products through simulation-based design

Assesses simulation credibility through verification and validation

Answer questions:

Which are crucial factors/parameters, how do they affect key metrics? (sensitivity)

How safe, reliable, robust, or variable is my system? (quantification of margins and uncertainty: QMU, UQ)

What is the best performing design or control? (optimization)

What models and parameters best match experimental data? (calibration)

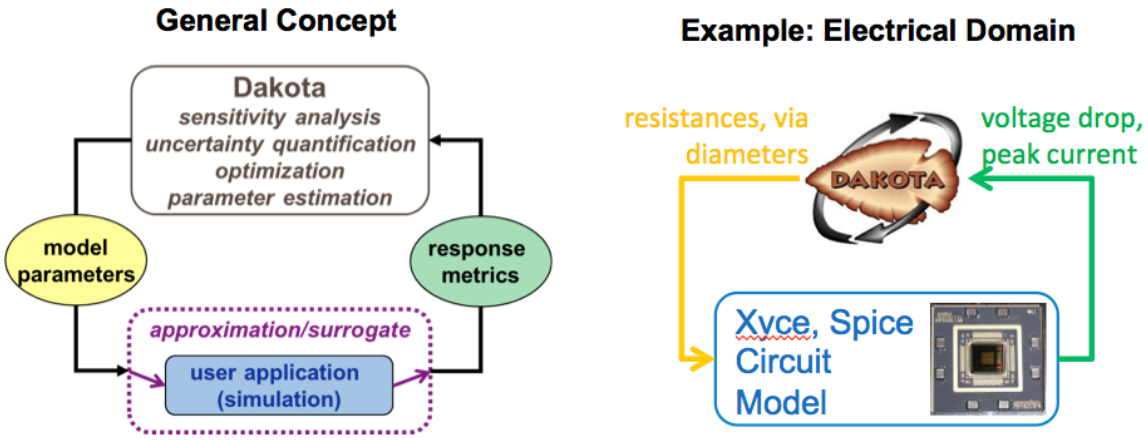


Figure 34. Dakota Framework Integration Wraps User Application

To put margins and uncertainty into context, assume that there is a device that is subject to heat, and we need assess some type of thermal uncertainty quantification. Given some results from some Design of Experiment (DoE) (also supported by Dakota) results that give a probability distribution as shown in Figure 35 [2]. The Mean of the temperature: T, to the lower bound of the threshold (e.g., 72 degrees) characterizes the Margin, and the Standard Deviation (T) characterizes the uncertainty.

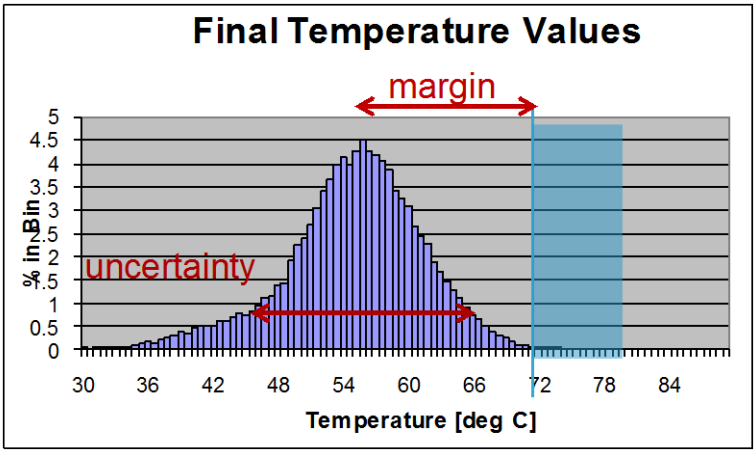


Figure 35. Example for Understanding Margins and Uncertainty

This approach and Dakota supports a broad set of domains, and therefore we think it can be generally applied across domain for NAVAIR, for example:

Supports simulation areas such as: mechanics, structures, shock, fluids, electrical, radiation, bio, chemistry, climate, infrastructure

Is best used with a goal-oriented strategy:

Find best performing design, scenario, or model agreement

Identify system designs with maximal performance

Determine operational settings to achieve goals

Minimize cost over system designs/operational settings

Identify best/worst case scenarios

Calibration: determine parameter values that maximize agreement between simulation and experiment

Handles parallelism, which is often not feasible with commercial tools, and why HPC can play an important role

Provides sensitivity analysis – find the most influential variables

Uncertainty Quantification

Models inherently have uncertainty

Assess effect of input parameter uncertainty on model outputs

- Determine mean or median performance of a system
- Assess variability in model response
- Find probability of reaching failure/success criteria (reliability)
- Assess range/intervals of possible outcomes

QUANTIFICATION OF MARGINS UNDER UNCERTAINTY

Dakota is a tool framework that can support the method of Quantification of Margins Under Uncertainty (QMU). Some of the material from Sandia is categorized "Official Use Only [OUO]." We provide a summary extracted from publically available information [66].

QMU pre-dates Dakota and is not unique to Sandia as it was used at Lawrence Livermore National Laboratory and Los Alamos National Laboratory, with the original focus of the methodology to support nuclear stockpile decision-making[21]. QMU is a physics package certification methodology and although it has been around and used at Sandia dating back to 2003, and both QMU theory and implementation are still being developed/evolved [66]. We believe the methodology has more general use than just physics package certification.

QMU applies to the lifecycle of the whole weapon, with focus on:

Specification of performance characteristics and their thresholds

- Performance is the ability of system/component to provide the proper function (e.g., timing, output, response to different environments) when exposed to the sequence of design environments and inputs

Identification and quantification of performance **margins**

- A performance margin is the difference between the required performance of a system and the demonstrated performance of a system, with a positive margin indicating that the expected performance exceeds the required performance

Quantification of uncertainty in the performance thresholds and the performance margins as well as in the larger framework of the decisions being contemplated

There are two types of uncertainty that are generally discussed that account for, quantify, and aggregate within QMU:

Aleatory uncertainty (variability)

- Variability in manufacturing processes, material composition, test conditions, and environmental factors, which lead to variability in component or system performance

Epistemic uncertainty (lack of knowledge)

- Models form uncertainty, both known and unknown unknowns in scenarios, and limited or poor-quality physical test data

The statistical tolerance interval methodology is an approach to quantification of margins and uncertainties for physical simulation data. There is also probability of frequency approach commonly used in computational simulation QMU applications [66], which:

[21] The Comprehensive Nuclear Test Ban Treaty ends full-scale nuclear weapons testing in the U.S. President Bill Clinton at the United Nations, September 24, 1996

Extends the “k-factor” QMU methodology for physical simulation data

- k-factor, in general, is defined as margin divided by uncertainty (M/U)
- Margin (M): difference between the best estimate and the **threshold** for a given metric
- Uncertainty (U): the range of potential values around a best estimate of a particular metric or threshold

  - Provides essential engineering analysis to ensure the collected data sample includes measurements that may be used to infer performance in actual use
  - It is important to understand the performance requirement to understand the **performance threshold and associated uncertainty**

- Threshold: a minimum or maximum allowable value of a given metric set by the responsible Laboratory

The new method addresses the situation where performance characteristic has shown the potential for low margin or a margin is changing (likely getting smaller or there is greater uncertainty) with age [66]

- Notionally the margin shifts from the mean of the performance characteristic (PC) and its performance requirement (PR) to the difference between a meaningful percentile of the distribution of the performance characteristic and its performance requirement
- Need to quantify uncertainty through the computation of a statistical confidence bound on the best estimate of the chosen percentile rather than by a sample standard deviation (as reflected in Figure 35), which does not account for sampling variability
- This is accomplished by computing a statistical tolerance interval

We created a graphic from several publically available sources, as shown Figure 36 in order to better explain a few aspects about QMU, Dakota, epistemic and aleatory uncertainty. Typically within the Dakota framework there is an outer loop: epistemic (interval) variables and inner loop: uncertainty quantification over aleatory (probability) variables (e.g., the probability distribution). The outer loop determines interval on statistics, (e.g., mean, variance). The inner loop uses sampling to determine the responses with respect to the aleatory variables. This information can be used to understand the epistemic and aleatory uncertainties, relative to the Lower Performance Requirement (LPR).

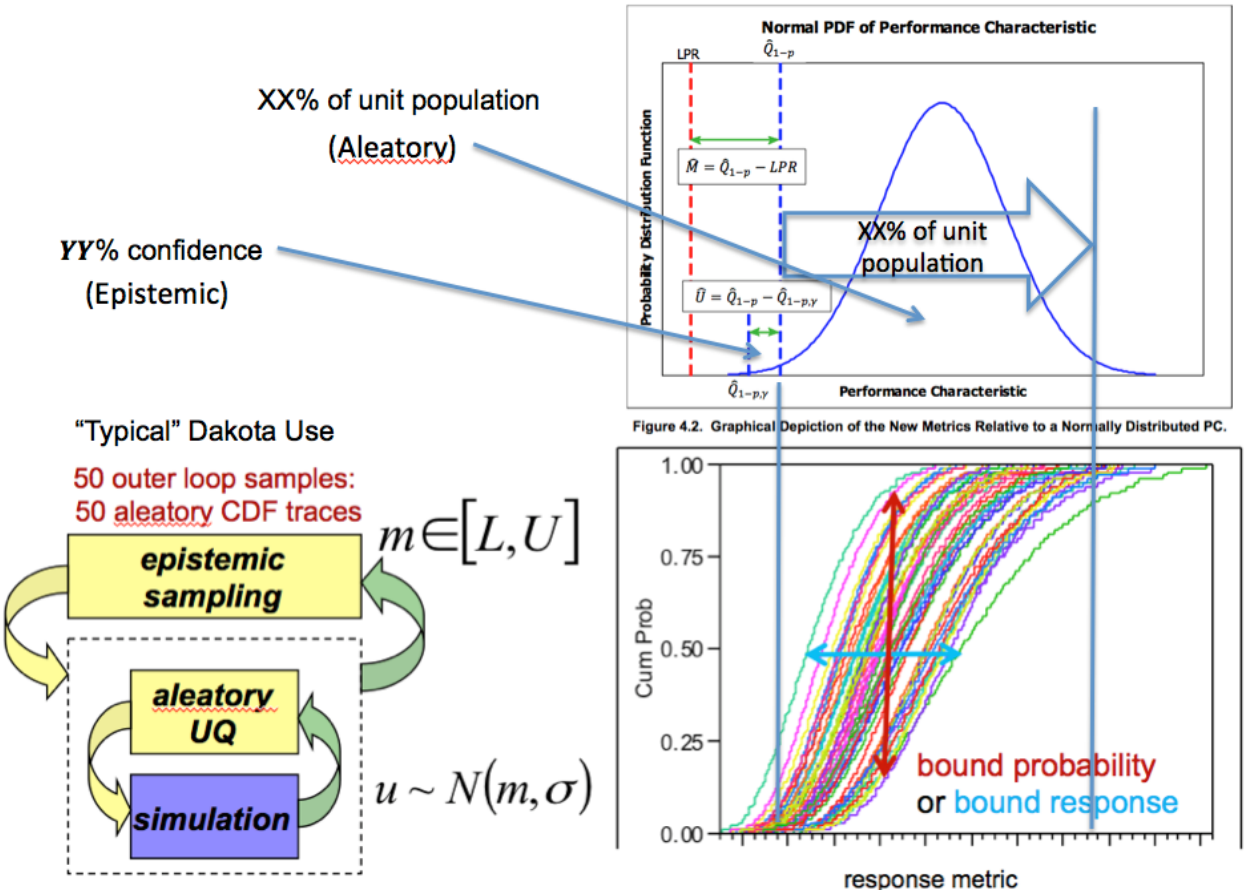


Figure 36. Pulling Together Concept Associated with QMU

The information is relevant to the risk framework as it provides evidence about methodologies and tools to deal with several of the topics discussed in Section 6.2. QMU and Dakota are still evolving, and there are a number of challenges:

How do we ensure that we use the right "data" as inputs?

How to roll up to the system level?

Model validation and simulation qualification (see Section 6.5)

RISK FRAMEWORK APPROACH TO UNCERTAINTY MODELING AND PREDICTION

The SERC team has also been working with NAVSEA to develop a framework and approach to uncertainty quantification modeling and prediction. The approach has three main components:

Identifying the design, test and modeling factors at different system scales
Analyzing the uncertainty, variability, and error in design implementation, testing, and modeling
Using experimental design methods to assess the contributions and interactions to system (airworthiness and safety) and program execution risks

The risk modeling and analysis approach also addresses potential errors and uncertainties in the overuse of limited data. Ideally:

One data set is used to identify critical factors

A second independent data set is used to develop the models

A third independent data set is used to calibrate the models

A fourth independent data set is used to assess the expected error in model results

In practice data sets, surrogate vehicle test data, etc. are limited. Bootstrap methods use repeated resampling of the data and repeating the modeling and analysis process to obtain a statistical estimate of the uncertainty in model-based acquisition given the available data. Further analysis reveals the value – reduction in uncertainty – for additional data.

These types of models capture and embed knowledge associated with expert judgment, historical evidence and rules of thumbs that are used in the decision-making process. Alternative methods such as those discussed in Section 6.4.1 help deal with these type of issues.

PREDICTIVE MODELS FOR RISK

There are situations where we do not have good historical quantitative data and we often use expert judgment. This section discussions a predictive modeling approach when risk involves subjective information, small data sets, and "dirty" data.

The SERC team has developed and used models in the prediction of risk, and plans to use predictive analytic models to support risk identification and management. More generally we can use models to provide risk quantification for almost all types of decisions that are made by stakeholders (e.g., model-based reviews). As an example, we created a Bayesian model using factors derived from the Airworthiness standard MIL-HDBK-516B [37] as shown in Figure 37. This is conceptually similar to the approach we are using on an FAA NextGen research task for collaborative risk-informed decision-making [10][11][12]. The key characteristics of the approach are they ensure that all factors are considered in the decision-making process, and that all classes of stakeholders are adequately represented in the decision-making process. A systematic and comprehensive treatment of all relevant factors provides better risk identification.

We used this model and an example from a true story related to a C130 Weapon Delivery system to illustrate the concept. While this model is notional at this time, this example started a discussion with the team about how stochastic (probabilistic) models can play an important part of the Vision as they formalize many aspects of the human decision making process that will be important at many gates, reviews, and decision points of the Vision concept. Each factor covers a specific aspect of airworthiness, to ensure that all possible uncertainties and risk are considered in the quantification of risk. The risk index is a probability distribution, where for example, the mean can map to quantities in a risk matrix.

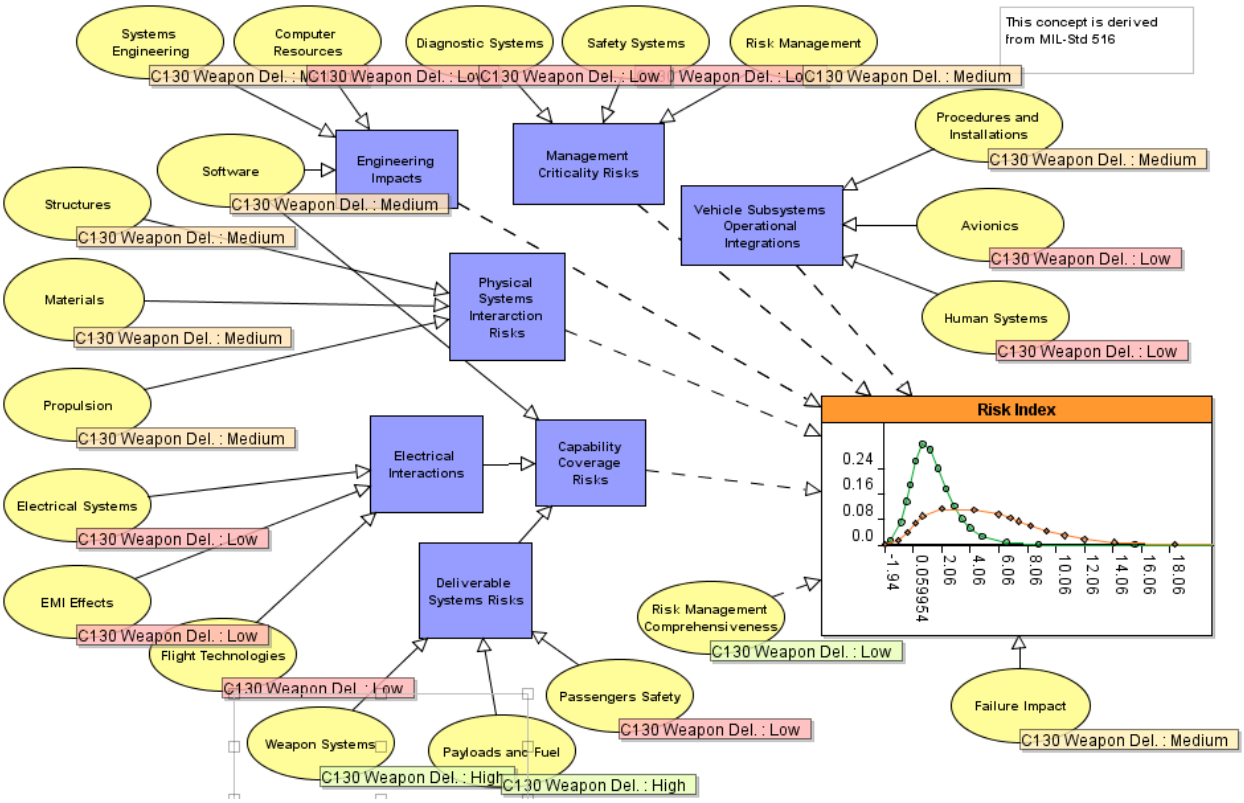


Figure 37. Bayesian Model Derived from Airworthiness Factors

RISK FRAMEWORK CAPTURES KNOWLEDGE

These types of risk frameworks are actually knowledge models of credibility (not models of performance, but models of uncertainty). Part of the effort on modeling the "As Is" process (Task 3) is to identify and then formalize within the models the information and associated knowledge for evidence-based decisions and evidence-based timing of decisions. Other considerations and opportunities:

In the "As Is" process, what decisions are artifacts of the process, but not essential to the engineering development?

Are there lost opportunities by making early concept and design decisions?

Is there a risk of bad decisions, risks and costs of no or deferred decisions, during planning, or during execution?

Reconsider the "full system" technical review model. Not all parts of the system are ready for PDR, CDR at the same time. Some are more mature than others. Maybe a granular approach is needed.

The timing of technical reviews and decisions should be made when there is an accumulation of evidence sufficient to make a credible decision. Ideally, this will be inherent in the Vision concept, when the required information and associated analyses are complete, the evidence and timing for decisions should be triggered events in the automated workflow.

Comparing model predictions to observed responses in this manner for the purpose of assessing the suitability of a particular model constitutes what is known as *model validation*. Uncertainty quantification for simulation models is not strictly limited to model validation. When experimental observations are available for validation assessment, analysts would often like to use the same observations for *model calibration*, which is the process of adjusting internal model parameters in order to improve the agreement between the model predictions and observations. But if internal model parameters are allowed to be adjusted in this manner, this means that there is some amount of uncertainty associated with the true, or best, values of these parameters. And uncertainty associated with model inputs directly implies uncertainty associated with model outputs [58].

Model validation and simulation qualification are ways to ensure that "integrity" of the models prediction information. Sandia has developed the "Real Space" model validation approach [77], which was formulated by working backwards from an end objective of "best estimate with uncertainty" (BEWU) modeling and prediction, where *model validation* is defined as: the process of determining the degree to which a computer model is an accurate representation of the real world from the perspective of an intended use of the model. However, the interpretational and implementation details can still vary widely.

We have discussed a number of model validation and simulation qualification topics, such as:

Hierarchical Model Validation

- Seeks to expose key physics and material models that are brought together, and asks are the combined products validated at various levels of aggregation? "right for the right reasons"
- Seeks to catch interactions and emergent behaviors not present in validation of separate models
- Also need to consider "Traveling" or "Linking" variables that bridge modeling levels [78]

"Exercising" the models at the "boundaries" of the probability distributions (~10 and 90 percentile)

- This is related to a recommended testing strategy based on boundary-value analysis/testing (i.e., exercising the "element under test" at the boundaries can expose more anomalies that exercising the nominal/typical tests scenarios)
- Has greater potential to expose off-nominal cases

Various model validation paradigms and methodologies are still being proposed, developed, and tested. There is no overriding consensus exists yet on "best" approach. We questioned Sandia about an idea that we had in our working session about how we are increasing in the ability to do more "integration" of the simulation across domains, and can that "integration" provide increased visibility into potential anomalies, therefore allowing us to better understand the "integrity" of the simulations. This is analogous to why integration testing often exposes issues

Sandia provided some papers that we can share with the team [78]. This information provides significant guidance and historical perspectives that should be further used to support the concept of model validation and model integrity as part the Vision for Task 3.

There is more research planned for a follow-on phase of this research. Here are other topics that have discussed related to improving our trust in models and simulation:

Numerical integration techniques [44]

- This is an example provide by NASA/JSC related to simulation of space vehicles for different planetary bodies
- Propagating the evolution of a vehicle's translational and/or rotational state over the course of a simulation is an essential part of every space-based Trick simulation. The underlying equations of motion for this state propagation yield second order initial value problems. While analytic solutions do exist for a limited set of such problems, the complex and unpredictable nature of the forces and torques acting on a space vehicle precludes the use of analytic methods for a generic solution to these state propagation problems. Numerical integration techniques must be used to solve the problem.

Flights validate models/simulations

Use logged data to continually calibrate models/simulation

- We heard this discussed in our organizational visits, and it was discussed as part of model guidance
- Model calibration should be getting easier, because we have better data collection, storage, and the ability to analyze large data sets

Models of pedigree

Discussed the need for a new concept, a Model Validation Review (MVR)

Cross-domain integration of models may also be a way to have greater confidence in simulation models

- We know that integration and integration testing often exposes many defects or anomalies
- We currently do not have much cross-domain integration of models/simulation
- These are new capabilities and the inherent nature of model-centricity will lead to greater integration; this could potentially provide new types of inputs/measures (insights) to help us build trust in the models

Probabilistic Risk Analysis – this might be yet another related cross-domain approach

- Organization discussed an example related to using simulation and Dakota to reduce the number of flight tests

Bayesian model calibration [57]

- Model calibration is a particular type of inverse problem in which one is interested in finding values for a set of computer model inputs, which result in outputs that agree well with observed data

Finally, Bill Brickner from NAVAIR points out that no mission model can ever be validated – that is, it is being used to predict possible future scenarios. We will continue to investigate approaches.

RISK IN A COLLABORATIVE ENVIRONMENT

Risk is not limited just to NAVAIR, it must be considered during the interactions with contractors in a continuous way rather than the monolithic reviews, especially in the context of a "radical transformation." For example, can we create a means to enable NAVAIR to continuously use model measures as an assessment of the design and risk of a continuously evolving contractor's design/system rather than having document-based reviews? If so, then:

There might be a need for new types of policies

It has been suggested that there may need to be some type of a policy reference model that:

Provides a common way to guide the use of artifacts to make decisions

Identify evidence (derived from models)

Access information

Analyze information leading to a Decision

- With quantification of uncertainties

Could be related to new SETR Manager and this builds on the discussion we had about Airworthiness/EDRAP Concepts

Some "radical" transformation thoughts

A continual assessment of the model (all of the models) maturity

- Possibly with different Model Maturity Levels (MML)

If the models cover all aspects of the aircraft

- Related to the reference architecture/model of an aircraft

We can have a cumulative quantity that represents the state of the design and a measure of risk (uncertainty relative to our understanding of the margins)

It must be stated that the above scenarios about "eliminating reviews" through the use of model measures does not eliminate the interaction between NAVAIR and contractors, rather we suggest that there is a need for continuous collaboration among all stakeholders and those interactions can be done on a weekly basis or in a more workshop-based approach in the context of models.

In this approach, could a “radical transformation” in the way that government and contractors interact reduce risk? One of the organizational discussions reflected on this concept of continuous collaboration using model; the following is a true and positive story related to a Navy customer in the use of Simulink modeling:

The contractor started this interaction, because they were several years behind, and had not made any “real” progress

They started modeling, which uncovered many requirement errors/issues

This helped them understand the complexity and realized the effort was much more extensive than they had originally estimated

Started open discusses with their customer (Navy)

- Models provided tangible technical information about the problem
- After a little explanation about the modeling approach, the customer was able to understand the models
- They both realized that requirement shall statements cannot provide the needed information, and many were just wrong (incorrect, contradictory, or not what the Navy wanted)
- Documented issues directly in the models
- Realized that the models were in a constantly changing state, but the contractor built a trust relationship with the Navy understanding that the models were in a continuously evolving state
- Each passing week they would review the models and could reflect on the issues that were recorded in the models

There are other variants of the operational model that were recently discussed by NASA/JPL in the way that they use models and reviews in a different way that the traditional “gate” reviews (e.g., SRR, SFR) in a model-centric way [28].

RISK RELATED RESEARCH

SERC research teams are involved in several related research efforts that will be leveraged in the risk framework. We need to explore how the following can be leveraged:

Trust under Uncertainty - Quantitative Risk; SERC RT-107 [87].

The High Performance Computing Modernization (HPCM) CREATE program to use high-fidelity models in systems design is establishing a working group on Uncertainty Quantification. SERC partners are collaborating with NAVSEA and the HPCM program.

The DARPA internet-fabrication (iFab) project sponsored research by a SERC collaborator to develop software to automatically detect and complete gaps in specifications for a “build to” design.

The US Army TARDEC is developing knowledge models to capture design factors and relationships in system design and development. The resulting decision breakdown structure

and process should help distinguish substantive design and engineering decisions versus artifacts of the “As Is” process. SERC partners are coordinating with this effort.

OSD is sponsoring a SERC project in “risk leading indicators” and “risk estimating relationships,” analyzing the consistency, completeness, and complexity of the system architecture, requirements, task structure, and team organization, and combining those with TRL/IRL levels and Advancement Degree of Difficulty indicators (this project is being conducted in collaboration with TARDEC and an acquisition program).

The Engineered Resilient Systems (ERS) effort is addressing lost opportunity by making early concept & design decisions, the time and cost to reverse decisions, and tradeoffs between timely but bad decisions versus deferred decisions. SERC partners are collaborating with the NAVSEA ERS and set-based design projects.

SUMMARY AND NEXT STEPS

We have conducted over 29 discussions, including 21 on site, as well as several follow-up discussions on some of the identified challenge areas. Our research suggests that model-centric engineering is in use and adoption seems to be accelerating. Model-centric engineering can be characterized as an overarching digital approach for integrating different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity across disciplines throughout the lifecycle. We seem to be getting closer to a tipping point where we are progressing beyond model-based to model-centric where integration of computational capabilities, models, software, hardware, platforms, and humans-in-the-loop allows us to assess system designs using dynamic models and surrogates to support continuous and often virtual verification and validation in the face of changing mission needs.

Enabling digital technologies are changing how organizations are conceptualizing, architecting, designing, developing, producing, and sustaining systems and systems of systems (SoS). There are many enablers that relate to characteristics of a holistic approach to model-centric engineering such as (this list is not exhaustive):

Mission-level simulations that are being integrated with system simulation, digital assets & products providing a new world of services

Leaders are embracing change and adapting to use digital strategies faster than others

Modeling environments to create dynamic operational views (e.g., DoDAF OV-1) are increasingly used, which used to be static pictures

1D, 2D & 3D models have simulation and analysis capabilities (mostly physics-based) are common in practice

Platform-based approaches with virtual integration help automakers deliver vehicle faster

Modeling and simulation in the automotive domain is reducing the physical crash testing (e.g., from 400 to 40); this could imply that modeling and simulation can reduce test flights, which are very costly as it is difficult to get flight clearances on air craft that have advanced new capabilities

Design optimization and trade study analysis allows for more systematic design of experiments and allows engineering to make many more excursions through the design space

Engineering affordability analysis is a risk-based approach that could be used to significantly reduce flight tests by focusing on those flights that have the most uncertainty about margins of performance

Risk modeling and analysis

Pattern-based modeling based on ontologies with model transformation and analysis

Domain-specific modeling languages

Set-based design

Modeling and simulation of manufacturing

Our discussion also identified some challenge areas, such as:

The growth and complexity of software is an increasing challenge especially given the fact that 90 percent of the functionality in a 5th generation air vehicle system is in software; in addition due to the needs for airworthiness and safety, software verification is critical, but often results in longer than expected durations and schedule slips

There is an "explosion of models," however, there is a lack of cross-domain model interoperability, consistency, and limitations transforming models with the required semantic precision to provide accurate information for decision making

Unvalidated models as the cost for verification, validation and accreditation can be expensive and this can lead to incorrect or invalid results leading to organizations not identifying design or integration problems until late in the lifecycle

This list is not exhaustive. This report provides some scenarios about how to address the first item, and we will have some follow-up discussions with organizations, and further investigate root causes, which might be addressed by early modeling and simulation to produce "better" requirements. We also think the second item can be addressed through "engineering," and NAVAIR is making some headway on this item. The third topic relates to a question posed by our sponsor after our review of the material presented in this report, paraphrased:

If we are going to rely more heavily on model-centric engineering, with an increasing use of modeling and simulations, how do we know that models/simulations used to assess "performance" have the needed "**integrity**" to ensure that the performance predictions are accurate (i.e., that we can trust the models)?

Our visit to Sandia National Laboratory and one industry organization provided some insights into model-centric approaches and tools they are using that can address aspects of this topic. We believe that their approach and tools provide a measure of certainty into a model's predictive capabilities, and measures of uncertainty of these predictive capabilities can apply to almost any model/simulation.

Model-centric engineering technologies enable more automation and efficiencies, however while research suggests that it is technically feasible to create a holistic approach for conceiving innovative concepts and solutions enabled through model-centricity, our sponsor is looking for a radical transformation to change how we operate to coordinate the efforts across multiple disciplines with all relevant stakeholders at the right time and virtually. The concept for a radical transformation still needs to be addressed as we move forward.

There are also some emerging ideas that will impact the Vision model and "end state," for examples:

Computer augmentation, where digital assistance will begin to understand what we are trying to model and through advances such as machine learning and integrated visualization can act as a knowledge librarian helping us to model some aspects of the problem or solution at an accelerating pace

Ontologies used in new ways to bridge the gap on model semantics mismatch and compositional views across domains

Explosion of interactive visualization, which we will need as we have a "sea" of data and information derived from a "sea" of models with HPC computing capabilities limited only by our ability to cool those systems

Our research finding address most aspects of the technical feasibility research questions. We did develop some scenarios where we can argue that it is technical feasible to achieve a 25 percent reduction in time to develop large scale air vehicle systems enabled by model-centric engineering. However, we need to further investigate the feasibility of the scenarios through follow-up discussion. In addition, we have identified challenges and gaps. We believe the following research questions, some of which come directly as follow-up questions from our sponsor, must be addressed going forward:

How do we address the gaps and challenges identified during the organizational discussions, some of which are summarized in Section 2.5?

- We will continue to have follow-on discussions with a number of the organizations that we have visited (Task 1), and we are planning an Industry Day on model-centric engineering

What is the Vision, how do we represent it, how do we characterize an "end state," how does it relate to a radical transformation, and how does such a Vision relate to the As Is and Airworthiness process?

- Define the interface boundaries for mission-level integration as it relates to model-centric engineering at the program of record level

If we are going to rely more heavily on model-centric engineering, with an increasing use of modeling and simulations, how do we know that models/simulations used to assess "performance" have the needed "integrity" to ensure that the performance predictions are accurate?

- Clarify how to systematically address model verification, validation, and simulation qualification (ensure model "integrity" with trust)
- Term used by our sponsor, with the implied meaning that we have trust and/or evidence in the accuracy of model's predictive capabilities

How do we integrate a risk framework into the Vision and "end state" to support risk-informed decision-making in a world characterized through model-centricity?

Can model-centric engineering enable a radical transformation in the way that NAVAIR operates?

- We need to characterize a "radical transformation" and the associated "end state." With the context of a Vision model, we need to explore alternative operational concepts such as decision frameworks that were identified through the organizational meetings

To the extent possible, we need to:

Discuss the content and representation of the lexicon (Task 2) that has been transferred to NAVAIR

Test out a vision representation concept with our sponsors

Define a timeline/roadmap for the Vision for addressing the challenge areas

Demonstration thread(s) based on case study or surrogate data

APPENDIX A: FACTOR DEFINITIONS

The following is the current set (fifth version) of the set of factors associated with the discussion measurement instrument (see Section 2). As the discussions with organizations are held, these factors and the associated categories will be refined.

Table 3. Discussion Instrument Factor Definition

Factor Category	Factors	General These factors relate to the degree to which advance MBSE provides a holistic approach to SE	Commentary
Magnitude of applicability over the lifecycle	Organizational Scope	What is the scope of the MBSE usage? Normally, when thinking about NAVAIR systems the scope is quite large and involves large programs. Therefore, what organizational scope does the MBSE usages apply: Program, Project, an entire Business Unit, Platform (e.g., a type of a specific aircraft, tank, automobile), Department, or Site.	Key to all of these questions is that we are looking for "Technical Feasibility of "doing everything with model." We recognize that actual adoption can be difficult, and might not make sense on older systems. Therefore related to this, it is probably best to ensure that the question perspectives come from a Chief Engineer, Chief Technical Offer, MBSE Organizational Specialist and possibly the Program Manager. To carry this a step further, it might also be important to keep the "systems" perspective in mind, because some of the concepts discussed may have been applied in Hardware and possibly in Software (e.g., the control laws for the F35 are built in Simulink, with auto code generation, and a significant portion of auto test generation), but not completely at the Systems level.
	Scope Impact	How broadly does the answers cover the entire lifecycle (for example, a university research project might be very advanced in terms of analysis or simulation, but it does not cover the entire DoD 5000 lifecycle).	The answer to this question has a lot of weight, because we need to consider answer in context of lifecycle applicable to NAVAIR (and in general DoD Acq. Programs).
Proven beyond a research concept	Demonstrations	Are the capabilities discussed actually in operations - have they been demonstrated?	We want to understand that things discussed are more than just research concepts.
Crossing the Virtual V	Integrated Simulation	In order to Cross the Virtual V, there will be many types of modeling and simulation required to support various type of domains within the system. To what degree are the simulations integrated, and better yet do different simulations work off of shared models?	In order to "cross the virtual V" during the early stages of development, it is important to understand if the inputs/outputs from one set of simulations can feed another (e.g., to be able to understand the capability in the mission context)
	Formal Analysis	Are the analyses (e.g., property analysis) formal, meaning that they are performed on models automatically?	Is the analysis fully automated from the models (H) or is there human interpretation required (M or L) or none (L)?

	Domain Specific	Are the different types of models related to the domain? For example, control system engineers often use Simulink/Matlab. Also, most modeling and simulation environments are domain-specific.	Domain-specific modeling languages are an emerging trend; these types of approaches provide intuitive abstractions (often graphical) that are familiar to engineers within the domain. Rather, SysML, while good for systems engineers, it might not be applicable to flight controls, networks, fluid dynamics, etc. In addition, there is not significant support for automated V&V from SysML as the semantics are not very rich.
Cross Domain Coverage	Domain Interoperability	Are the models that are in different, but related domains integrated? Are the models consistent across the domains?	For example, are the models that are used for performance the same models used for integrity/dependability analysis?
	Synthesis/Generation	Can the models be used for synthesis/generation of other related artifacts such as code, simulation, analysis, tests and documentation	
	Meta-Model/Model Transformations	Are the models used in one domain, or for one purpose, transformable into another domain where the well-defined semantics in one domain is carried through the transformation into the other domain; if so are they known to be consistent?	We know that one type of modeling is not always appropriate for everything, and that is why there is emergence of Domain-Specific Modeling languages; the key question is: are the models for one use consistent for other users (e.g., performance, integrity).
Virtual System Representation	Surrogate Integration	Are surrogates used to support analysis, and are the results of the surrogates captured so that they can be factored into modeling and simulation in the future?	Example, Formula 1 racing, uses data logging during physical experimentation and then factors results (and logs) back into simulation environment; can we fly some new capability on an existing aircraft and then factor the results from the test flights back into the modeling and simulation environments? This in the future should allow more virtual flight testing (once the approach becomes trusted).
	Formal Capability Assessment	How well do the models, simulations and analyses capabilities support the ability to understand the capabilities being developed?	Are the abstractions from the models still "rich enough" to be representative of the actual mission environment when used in a virtual environment? For example, if we use a 3D immersive environment, can we understand the physical characteristic of the operational system?
	Virtual Accuracy/Margin Analysis	Are the modeling, simulation and analysis accurate? How well do they allow the designers to understand the margins?	As an example, margin tolerances at the component level can propagate as the system is composed (or assembled). Are these factors understood and controlled?
	3D Immersive Environments	What is the degree to which 3D Immersive Environments are used to improve the understanding (and possibly training) of the virtual systems.	

Management Criticality Risks (Relates to Task 4)	Risk Management	Is there proper risk management identification, analysis and mitigations applied based on the use of models?	This should also consider: - Adequately deal with critical timelines - Integrated operational risk - Change management (model-based change management is different than document-based)
	Predictive Analytics	Are there models used to support a quantitative approach to risk management?	
Attributes of Modeling Maturity	Model-based metrics	Are there model-based metrics (or a comprehensive set of model measurements) and are they used to support the management of programs/projects?	The use of model-based metrics reflects on the modeling maturing of the organization.
	Multi-model interdependencies / consistency and semantic precision	If the organization is dealing with many different types of models, are the interdependencies managed and are the models semantically precise enough to manage model consistency?	Dealing with interdependencies and modeling consistency often deals with having a detailed understanding of the semantics across models. Positive results for this answer suggest a very advanced use of models.
	High Performance Computing (HPC)	Is HPC applied to the modeling, simulation and analysis efforts?	Use of HPC is an indicator of high modeling maturity.
Operational Risks (Relates to Task 4)	Procedures	Are the procedures for using the models understood, so that we can trust the model outputs to support other related types of analysis, both in terms of technical as well as risk?	This applies heavily in airworthiness (e.g., Mil Std. 516)
	Staff and Training	With the advances in the technologies associated with models, are the staff and training in place to support the use of models?	This is another indicator of a more advanced organization. As a side effect the use of 3D Immersive systems can be valuable in both collaboration and early training.
	Human Factors	How well are human factors supported by the modeling, simulation and analysis capabilities? This should consider Usability.	
Indirect support from Models	Certification	How well do the models-based automation and practices support certifications (if required)?	If not applicable use M - for Medium
	Regulation	How well do the models-based automation and practices support regulations (if required)?	If not applicable use M - for Medium
	Modeling and Simulation Qualification	How much do we trust our models?	

APPENDIX B: ACRONYMS AND ABBREVIATION

This section provides a list of some of the terms used throughout the paper. The model lexicon should have all of these terms and many others.

AADL	Architecture Analysis & Design Language
ACAT	Acquisition Category
AFT	Architecture Framework Tool of NASA/JPL
AGI	Analytical Graphics, Inc.
AGM	Acquisition Guidance Model
ANSI	American National Standards Institute
AP233	Application Protocol 233
ATL	ATLAS Transformation Language
ASR	Alternative System Review
AVSI	Aerospace Vehicle Systems Institute
BDD	SysML Block Definition Diagram
BN	Bayesian Network
BNF	Backus Naur Form
BOM	Bill of Material
BPML	Business Process Modeling Language
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CDR	Critical Design Review
CEO	Chief Executive Officer
CESUN	International Engineering Systems Symposium
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CORBA	Common Object Requesting Broker Architecture
CREATE	Computational Research and Engineering for Acquisition Tools and Environments
CWM	Common Warehouse Metamodel
dB	Decibel
DBMS	Database Management System
DAG	Defense Acquisition Guidebook
DARPA	Defense Advanced Research Project Agency
DAU	Defense Acquisition University
DCDR	Digital design from Critical Design Review (CDR)
DL	Descriptive Logic
DoD	Department of Defense
DoDAF	Department of Defense Architectural Framework
DoE	Design of Experiments
DSL	Domain Specific Languages
DSM	Domain Specific Modeling
DSML	Domain Specific Modeling Language
E/DRAP	Engineering Data Requirements Agreement Plan
ERS	Engineered Resilient Systems
FAA	Federal Aviation Administration
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
GAO	Government Accounting Office

HPC	High Performance Computing
HPCM	High Performance Computing Modernization
HW	Hardware
I&I	Integration and Interoperability
IBM	International Business Machines
IBD	SysML Internal Block Diagram
ICD	Interface Control Document
ICTB	Integrated Capability Technical Baseline
IDEF0	Icam DEFinition for Function Modeling
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council on Systems Engineering
IPR	Integration Problem Report
IRL	Integration Readiness Level
ISEF	Integrated System Engineering Framework developed by Army's TARDEC
ISO	International Organization for Standardization
IT	Information Technology
IWC	Integrated Warfighter Capability
JEO	Jupiter Europa Orbiter project at NASA/JPL
JSF	Joint Strike Fighter
JPL	Jet Propulsion Laboratory of NASA
Linux	An operating system created by Linus Torvalds
LOC	Lines of Code
M&S	Modeling and Simulation
MARTE	Modeling and Analysis of Real Time Embedded systems
MATRIXx	Product family for model-based control system design produced by National Instruments; Similar to Simulink
MBEE	Model-based Engineering Environment
MBSE	Model-based System Engineering
MBT	Model Based Testing
MC/DC	Modified Condition/Decision
MCE	Model-centric engineering
MDA®	Model Driven Architecture®
MDD™	Model Driven Development
MDE	Model Driven Engineering
MDSD	Model Driven Software Development
MDSE	Model Driven Software Engineering
MIC	Model Integrated Computing
MMM	Modeling Maturity Model
MoDAF	United Kingdom Ministry of Defence Architectural Framework
MOE	Measure of Effectiveness
MOF	Meta Object Facility
MOP	Measure of Performance
MVS	Multiple Virtual Storage
NASA	National Aeronautics and Space Administration
NAVAIR	U.S. Navy Naval Air Systems Command
NAVSEA	U.S. Naval Sea Systems Command
NDA	Non-disclosure Agreement
NDIA	National Defense Industrial Association
NEAR	Naval Enterprise Architecture Repository

NPS	Naval Postgraduate School
OCL	Object Constraint Language
OMG	Object Management Group
OO	Object oriented
OSD	Office of the Secretary of Defense
OSLC	Open Services for Lifecycle Collaboration
OV1	Operational View 1 – type of DoDAF diagram
OWL	Web Ontology Language
PDM	Product Data Management
PDR	Preliminary Design Review
PES	Physical Exchange Specification
PIA	Proprietary Information Agreement
PIM	Platform Independent Model
PLM	Product Lifecycle Management
POR	Program of Record
PRR	Production Readiness Review
PSM	Platform Specific Model
QMU	Quantification of Margins under Uncertainty
RT	Research Task
RFP	Request for Proposal
ROI	Return On Investment
SAVI	System Architecture Virtual Integration
SE	System Engineering
SERC	Systems Engineering Research Center
SETR	System Engineering Technical Review
Simulink/Stateflow	Product family for model-based control system produced by The Mathworks
SCR	Software Cost Reduction
SDD	Software Design Document
SE	System Engineering
SFR	System Functional Review
SLOC	Software Lines of Code
SME	Subject Matter Expert
SOAP	A protocol for exchanging XML-based messages – originally stood for Simple Object Access Protocol
SoS	System of System
Software Factory	Term used by Microsoft
SRR	System Requirements Review
SRS	Software Requirement Specification
STOVL	Short takeoff and vertical landing
SVR	System Verification Review
SW	Software
SysML	System Modeling Language
TARDEC	US Army Tank Automotive Research
TBD	To Be Determined
TRL	Technology Readiness Level
TRR	Test Readiness Review
UML	Unified Modeling Language
XMI	XML Metadata Interchange
XML	eXtensible Markup Language

US	United States
XSLT	eXtensible Stylesheet Language family (XSL) Transformation
xUML	Executable UML
Unix	An operating system with trademark held by the Open Group
UQ	Uncertainty Quantification
VHDL	Verilog Hardware Description Language
V&V	Verification and Validation
VxWorks	Operating system designed for embedded systems and owned by WindRiver

APPENDIX C: TRADEMARKS

Analysis Server is a registered trademark of Phoenix Integration, Inc.
Astah SysML is Copyright of Change Vision, Inc.
BridgePoint is a registered trademark of Mentor Graphics.
Cameo Simulation Toolkit is a registered trademark of No Magic, Inc.
CORE is a registered trademark of Vitech Corporation.
IBM™ is a trademark of the IBM Corporation
iGrafx is a registered trademark of iGrafx, LCC.
Java™ and J2EE™ are trademark of SUN Microsystems
Java is trademarked by Sun Microsystems, Inc.
LDRA is a registered trademark of Trademark of LDRA Ltd. and Subsidiaries.
Linux is a registered trademark of Linux Mark Institute.
Mathworks, Simulink, and Stateflow are registered trademarks of The Mathworks, Inc.
MagicDraw is a trademark of No Magic, Inc.
MATRIXx is a registered trademark of National Instruments.
Microsoft®, Windows®, Windows NT®, Windows Server® and Windows VistaTM are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. ModelCenter, is a registered trademark of Phoenix Integration, Inc.
Modelica® is a registered trademark of the Modelica Association.
Object Management Group (OMG): OMG's Registered Trademarks include: MDA®, Model Driven Architecture®, UML®, CORBA®, CORBA Academy®, XMI®
OMG's Trademarks include, CWM™, Model Based Application Development™, MDD™, Model Based Development™, Model Based Management™, Model Based Programming™, Model Driven Application Development™, Model Driven Development™
Model Driven Programming™, Model Driven Systems™, OMG Interface Definition Language (IDL)™, Unified Modeling Language™, <<UML>>™
OMG®, MDA®, UML®, MOF®, XMI®, SysML™, BPML™ are registered trademarks or trademarks of the Object Management Group.
Oracle and Java are registered trademarks of Oracle, Inc. and/or its affiliates.
ParaMagic is a registered trademark of InterCAX, Inc.
PHX ModelCenter is a registered trademark of Phoenix Integration, Inc.
PowerPoint is a registered trademark of Microsoft, Inc.
Real-time Studio Professional is a registered trademark of ARTiSAN Software Tools, Inc.
Rhapsody is a registered trademark of Telelogic/IBM.
Rose XDE is a registered trademark of IBM.
SCADE is copyrighted to Esterel Technologies.
Simulink is a registered trademark of The MathWorks.
Stateflow is a registered trademark of The MathWorks.
Statemate is a registered trademark of Telelogic/IBM.
STK is a registered trademark of Analytical Graphics, Incorporated (AGI), Inc.
UNIX is a registered trademark of The Open Group.
VAPS is registered at eNGENUITY Technologies.

VectorCAST is a registered trademark of Vector Software, Inc.
Visio is a registered trademark of Microsoft, Inc.
VxWorks is a registered trademark of Wind River Systems, Inc.
Windows is a registered trademark of Microsoft Corporation in the United States and other countries.
XML™ is a trademark of W3C
All other trademarks belong to their respective organizations.

APPENDIX D: REFERENCES

[1] Ackoff, R., L, and Sheldon Rodin. Redesigning Society. Stanford: Stanford University Press, 2003.

[2] Adams, B., Adam Stephens, Dakota Sensitivity Analysis and Uncertainty Quantification, with Examples, SNL 6230 Course on UQ/SA, April 23, 2014.

[3] Allen, G., F. Hartman, F. Mullen, Dynamic Multi-level Modeling Framework, Results of the Feasibility Study, NDIA, October 2013.

[4] Baitch, L., Randall C. Smith, Physiological Correlates of Spatial Perceptual Discordance in a Virtual Environment, General Motors Research & Development Center Virtual Environments Laboratory.

[5] Bayer, Todd J., Matthew Bennett, Christopher L. Delp, Daniel Dvorak, J. Steven Jenkins, and Sanda Mandutianu. "Update - Concept of Operations for Integrated Model-Centric Engineering at JPL," 1–15. IEEE, 2011. doi:10.1109/AERO.2011.5747538.

[6] Bayer, Todd, Seung Chung, Bjorn Cole, Brian Cooke, Frank Dekens, Chris Delp, I. Gontijo, et al. "11.5.1 Early Formulation Model-Centric Engineering on NASA's Europa Mission Concept Study." *INCOSE International Symposium* 22, no. 1 (July 2012): 1695–1710. doi:10.1002/j.2334-5837.2012.tb01431.x.

[7] Bergenthal, J., Final Report on the Identification of Modeling and Simulation Capabilities by Acquisition Life Cycle Phases, Johns Hopkins University/Applied Physics Laboratory, 16th Annual Systems Engineering Conference, October, 2013.

[8] Bergenthal, J., J. Coolahan, Final Report on the Identification of Modeling and Simulation Capabilities by Acquisition Life Cycle Phases, NDIA Systems Engineering Division Meeting, February 2014.

[9] Blackburn, M.R., Model-Driven Verification and Validation, Safe & Secure Systems & Software Symposium, June, 15-17 2010. Modified from Paul Eremenko, META Novel Methods for Design & Verification of Complex Systems, December 22, 2009.

[10] Blackburn, M., A. Pyster, R. Dillon-Merrill, T. Zigh, R. Turner, Results from Applying a Modeling and Analysis Framework to an FAA NextGen System of Systems Program, NDIA, October, 2013.

[11] Blackburn, M., A. Pyster, R. Dillon-Merrill, T. Zigh, R. Turner, Modeling and Analysis Framework for Risk-Informed Decision Making for FAA NextGen, INCOSE, June 2013.

[12] Blackburn, M., A. Pyster, R. Dillon-Merrill, T. Zigh, R. Turner, Using Bayesian Networks for Modeling an Acquisition Decision-Making Process for the FAA NextGen Systems of Systems, NDIA, October, 2012.

[13] Blackburn, M., R. Busser, A. Nauman, and T. Morgan. "Life Cycle Integration Use of Model-Based Testing Tools," 2:10.D.4–1 – 10.D.4–13. IEEE, 2005. doi:10.1109/DASC.2005.1563402.

[14] Blackburn, M., R. Busser, H. Graves, Guidelines for Automated Analysis of System Models, Software Produtivity Consortium Technical Report, December, 2000.

[15] Blackburn, M., Cloutier, R., Hole, E., Witus, G., 2014. Introducing Model-Based Systems Engineering Transforming System Engineering through Model-Based Systems Engineering (Technical Report No. TR-044). Systems Engineering Research Center.

[16] Blackburn, Mark, Robert Cloutier, Eirik Hole, and Gary Witus. *Introducing Model-Based Systems Engineering Transforming System Engineering through Model-Based Systems Engineering*. Technical Report. Systems Engineering Research Center, March 31, 2014. http://www.sercuarc.org/library/view/58.

[17] Blackburn, M., P. Denno, Virtual Design and Verification of Cyber-physical Systems: Industrial Process Plant Design, Conference on Systems Engineering Research, March, 2014; http://dx.doi.org/10.1016/j.procs.2014.03.006.

[18] Blackburn, M., S. Kumar, Evolving Systems Engineering through Model Driven Functional Analysis, NDIA System Engineering Conference, October 2009.

[19] Bleakley, G., A. Lapping, A. Whitfield, Determining the Right Solution Using SysML and Model Based Systems Engineering, (MBSE) for Trade Studies, INCOSE International Symposium, June, 2011.

[20] Boehm, B., Software Cost Estimation with Cocomo II, Prentice Hall, 2000.

[21] Box, George E. P. *Empirical Model-Building and Response Surfaces*. Wiley Series in Probability and Mathematical Statistics. New York: Wiley, 1987.

[22] Brat, Guillaume, V & V of Flight-Critical Systems, NASA ARCS5 - Safe & Secure Systems & Software Symposium, June 2010.

[23] Broy, M., M. Feilkas, M. Herrmannsdoerfer, S. Merenda, and D. Ratiu. "Seamless Model-Based

Development: From Isolated Tools to Integrated Model Engineering Environments." *Proceedings of the IEEE* 98, no. 4 (April 2010): 526–45. doi:10.1109/JPROC.2009.2037771.

[24] Business Process Modeling Notation. Retrieved March 2010, from Wikipedia, The Free Encyclopedia: http://en.wikipedia.org/wiki/Business_Process_Modeling_Notation.

[25] Browne, D., R. Kempf, A. Hansena, M. O'Neal, W. Yates, Enabling Systems Modeling Language Authoring in a Collaborative Web-based Decision Support Tool, Conference on System Engineering Research (CSER), March, 2013.

[26] Castet, Jean-Francois, Matthew L. Rozek, Michel D. Ingham, Nicolas F. Rouquette, Seung H. Chung, J. Steven Jenkins, David A. Wagner, and Daniel L. Dvorak. "Ontology and Modeling Patterns for State-Based Behavior Representation." American Institute of Aeronautics and Astronautics, 2015. doi:10.2514/6.2015-1115.

[27] Chilenski, J., SAVI Principal Investigator, Don Ward, TEES SAVI Program Manager, NDIA M&S Subcommittee – Arlington, Virginia 8 April 2014.

[28] Cooke, B., MBSE on Europa Clipper, NASA/JPL Symposium and Workshop on Model-Based Systems Engineering, January 2015.

[29] Coolahan, J. A Vision for modeling and simulation at APL, Johns hopkins APL Technical Digest, Volume 26, number 4 (2005).

[30] Dahmann, J., BA. Aumber, M, Kelley, Importance of Systems Engineering in Early Acquisition, MITRE Corporation. Approved for Public Release; Distribution Unlimited Case # 09-0345.

[31] DARPA, Producible Adaptive Model-based Software (PAMS) technology to the development of safety critical flight control software. PAMS has been developed under the Defense Advanced Research Projects Agency (DARPA) Disruptive Manufacturing Technologies program. Contract # N00178-07-C-2011, http://www.isis.vanderbilt.edu/projects/PAMS.

[32] Darwiche, A., Modeling and Reasoning with Bayesian Networks, Cambridge University Press, 2009.

[33] Davidoff, S., Visualization of Model Content and Engineering Process, NASA/JPL Symposium and Workshop on Model-Based Systems Engineering, January 2015.

[34] Defense Acquisition University, Defense Acquisition Guidebook Chapter 4 – Systems Engineering, May 2013; https://acc.dau.mil/dag4.

[35] Delp, C., D. Lam, E. Fosse, and Cin-Young Lee. "Model Based Document and Report Generation for Systems Engineering," 1–11. IEEE, 2013. doi:10.1109/AERO.2013.6496926.

[36] Department of Defense, INSTRUCTION – INTERIM, NUMBER 5000.02 November 26, 2013.

[37] Department of Defense, MIL-HDBK-516B, Department Of Defense Handbook: Airworthiness Certification Criteria, Feb, 2008; http://www.everyspec.com/MIL-HDBK/MIL-HDBK-0500-0599/MIL-HDBK-516B_CHANGE-1_10217.

[38] Department of Defense, Risk Management Guide For Dod Acquisition, Sixth Edition, August, 2006.

[39] Elele, J.N., Assessing Risk Levels of Verification, Validation, and Accreditation of Models and Simulations, International Test and Evaluation Association (ITEA) Journal 2008.

[40] Firesmith, D., Are Your Requirements Complete?, Journal of Object Technology, Volume 4, no. 1 (January 2005), pp. 27-43, doi:10.5381/jot.2005.4.1.c3.

[41] Graf, L., Transitioning Systems Engineering Research into Programs and Practice, NDIA 17th SE Annual Conference 2014.

[42] GAO, Problems Completing Software Testing May Hinder Delivery of Expected Warfighting Capabilities, GAO-14-322: Published: Mar 24, 2014. Publicly Released: Mar 24, 2014.

[43] Graignic, Pascal, Thomas Vosgien, Marija Jankovic, Vincent Tuloup, Jennifer Berquet, and Nadège Troussier, Complex System Simulation: Proposition of a MBSE Framework for Design-Analysis Integration, *Procedia Computer Science* 16 (January 2013): 59–68. doi:10.1016/j.procs.2013.01.007.

[44] Hammen, D., G. Turner, JSC Engineering Orbital Dynamics Integration Model, National Aeronautics and Space Administration, December 2014.

[45] Hannapel, Shari, Nickolas Vlahopoulos, and David Singer. "Including Principles of Set-Based Design in Multidisciplinary Design Optimization." American Institute of Aeronautics and Astronautics, 2012. doi:10.2514/6.2012-5444.

[46] Henson Graves, H., S. Guest, J. Vermette, Y. Bijan, H. Banks, G. Whitehead, B. Ison, Air Vehicle Model-Based Design and Simulation Pilot, Lockheed Martin, 2009; available http://www.omgwiki.org/MBSE.

[47] Hutchinson, J., J. Whittle, M. Rouncefield, S. Kristoffersen, Empirical Assessment of MDE in Industry,

Proceedings of the 33rd International Conference on Software Engineering, 2011.
[48] IDEFØ, Computer Systems Laboratory of the National Institute of Standards and Technology (NIST), 1993.
[49] International Council on Systems Engineering (INCOSE), "MBSE initiative," January 2007; https://connect.incose.org/tb/MnT/mbseworkshop/.
[50] ISO/IEC 42010:2007, Systems and Software Engineering -- Architecture Description, 2007.
[51] Jackson, Ethan, and Janos Sztipanovits. "Formalizing the Structural Semantics of Domain-Specific Modeling Languages." *Software & Systems Modeling* 8, no. 4 (September 2009): 451–78. doi:10.1007/s10270-008-0105-0.
[52] Jenkins, J. S., N. Rouquette, Semantically-Rigorous systems engineering modeling using SysML and OWL, 5th International Workshop on Systems & Concurrent Engineering for Space Applications, Lisbon, Portugal, October 17-19, 2012.
[53] Khan, O., G. Dubos, J. Tirona, S. Standley, Model-Based Verification and Validation of the SMAP Uplink Processes, IEEE Aerospace Conference, 2013.
[54] Kim, H., Fried, D., Menegay, P., Connecting SysML Models with Engineering Analyses to Support Multidisciplinary System Development, American Institute of Aeronautics and Astronautics, 2012.
[55] Kim, H., Fried, D., Menegay, P., G. Soremekun, C. Oster, Application of Integrated Modeling and Analysis to Development of Complex Systems, Conference on Systems Engineering Research, 2013; http://dx.doi.org/10.1016/j.procs.2013.01.011.
[56] Leveson, N., A New Accident Model for Engineering Safer Systems, Safety Science, Vol.42,No.4, April 2004.
[57] Matei, I., C. Bock, SysML Extension for Dynamical System Simulation Tools, National Institute of Standards and Technology, NISTIR 7888, http://dx.doi.org/10.6028/NIST.IR.7888, October 2012, http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7888.pdf.
[58] McFarland, J., Uncertainty Analysis For Computer Simulations Through Validation And Calibration, Dissertation, Vanderbilt University, May 2008.
[59] McFarland, J., Sankaran Mahadevan, Vicente Romero, Laura Swiler, Calibration and Uncertainty Analysis for Computer Simulations with Multivariate Output, AIAA, October, 2007.
[60] McKelvin, Jr., Mark, and Alejandro Jimenez. "Specification and Design of Electrical Flight System Architectures with SysML." American Institute of Aeronautics and Astronautics, 2012. doi:10.2514/6.2012-2534.
[61] MIL-HDBK-516C, Department Of Defense Handbook: Airworthiness Certification Criteria, December, 12, 2014.
[62] Murray, Brian T., Alessandro Pinto, Randy Skelding, Olivier L. de Weck, Haifeng Zhu, Sujit Nair, Narek Shougarian, Kaushik Sinha, Shaunak Bodardikar, and Larry Zeidner. *META II Complex Systems Design and Analysis (CODA)*, 2011.
[63] NAOMI Project, Lockheed Martin Advanced Technology Laboratories; http://www.atl.external.lmco.com/programs/STI/programs/program1.php#experimentalinfrastructure, 2013.
[64] NAVAIRINST 13034.1C, Navair Instruction: Flight Clearance Policy For Air Vehicles And Aircraft Systems, September, 28, 2004.
[65] Navy Integration and Interoperability (I&I) Integrated Capability Framework (ICF), Operational Concept Document, Version 2.0, 30 September 2013.
[66] Newcomer, J. T., SANDIA REPORT, SAND2012-7912 Unlimited Release Printed September 2012, A New Approach to Quantification of Margins and Uncertainties for Physical Simulation Data. (http://prod.sandia.gov/techlib/access-control.cgi/2012/127912.pdf).
[67] Nixon, D. W., Flight Control Law Development for the F-35 Joint Strike Fighter, October 5, 2004.
[68] Object Management Group, MBSE Wiki, Ontology Action Team, http://www.omgwiki.org/MBSE/doku.php?id=mbse:ontology, 2014.
[69] Object Management Group, XML Metadata Interchange (XMI), Version, 2.4.2, April 2014, http://www.omg.org/spec/XMI/2.4.2.
[70] Object Management Group. OMG Unified Modeling Language™ (OMG UML), Superstructure. 2011. Version 2.4.1. Available from: http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF.
[71] Object Management Group. OMG Systems Modeling Language (OMG SysML™). 2012. Version1.3. Available from: http://www.omg.org/spec/SysML/1.3/PDF.

[72] Paredis, C., Y. Bernard, R. Burkhart, D. Koning, S. Friedenthal, P. Fritzson, N. Rouquette, W. Schamai, An Overview of the SysML-Modelica Transformation Specification, INCOSE International Symposium, Chicago, IL, July, 2010.

[73] Pearl, J. (1985). "Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning" (UCLA Technical Report CSD-850017). Proceedings of the 7th Conference of the Cognitive Science Society, University of California, Irvine, CA. pp. 329–334. Retrieved 2009-05-01.

[74] Ray, S., G. Karsai, K. McNeil, Model-Based Adaptation of Flight-Critical Systems, Digital Avionics Systems Conference, 2009.

[75] Rasumussen, R., R. Shishko, Jupiter Europa Orbiter Architecture Definition Process, INCOSE Conference on Systems Engineering Research, Redondo Beach, California, April 14-16, 2011.

[76] Rhodes, D. H., A. M. Ross, P. Grogan, O. de Weck, [illegible] Engineering (IMCSE), Phase One Technical Report SERC-2014-TR-048-1, Systems Engineering Research Center, DSeptember 30, 2014.

[77] Romero, V., Elements of a Pragmatic Approach for dealing with Bias and Uncertainty in Experiments through Predictions: Experiment Design and Data Conditioning, "Real Space" Model Validation and Conditioning, Hierarchical Modeling and Extrapolative Prediction, SAND2011-7342 Unlimited Release Printed November 2011.

[78] Romero, V., Uncertainty Quantification and Sensitivity Analysis—Some Fundamental Concepts, Terminology, Definitions, and Relationships, UQ/SA section of invited paper for AIAA SciTech2015 Non-Deterministic Approaches Conference, Jan 5-9, 2015, Orlando, FL.

[79] Rothenberg, J. L. E. Widman, K. A. Loparo, N. R. Nielsen, The Nature of Modeling, Artificial Intelligence, Simulation and Modeling, 1989.

[80] Sandia National Laboratory, Dakota, https://dakota.sandia.gov/.

[81] Simko, Gabor, Tihamer Levendovszky, Sandeep Neema, Ethan Jackson, Ted Bapty, Joseph Porter, and Janos Sztipanovits. "Foundation for Model Integration: Semantic Backplane," 2012.

[82] Singer, David J., Norbert Doerry, and Michael E. Buckley. "What Is Set-Based Design?: What Is Set-Based Design?" *Naval Engineers Journal* 121, no. 4 (October 2009): 31–43. doi:10.1111/j.1559-3584.2009.00226.x.

[83] Spangelo, S. D. Kaslow, C. Delp, L. Anderson, B. Cole, E. Foyse, L. Cheng, R. Yntema, M. Bajaj, G. Soremekum, J. Cutler, MBSE Challenge Team, Model Based Systems Engineering (MBSE) Applied to Radio Aurora Explorer (RAX) CubeSat Mission Operational Scenarios, IEEEAC Paper #2170, Version 1, Updated 29/01/2013.

[84] Umpfenbach, E., Integrated System Engineering Framework (ISEF), NDIA Systems Engineering Conference, Oct. 2014.

[85] Wagner, D.A., M. Bennett, R. Karban, N. Rouquette, S. Jenkins, M. Ingham, An Ontology for State Analysis: Formalizing the Mapping to SysML, IEEE Aerospace Conference, 2012.

[86] Wikipedia, Ontology, http://en.wikipedia.org/wiki/Ontology_(information_science), 2014.

[87] Witus, G., W. Bryzik, Trust under Uncertainty - Quantitative Risk, SERC RT-107, Systems Engineering Research Review, December, 2014.

[88] Witherell, Paul, Boonserm Kulvatunyou, and Sudarsan Rachuri. "Towards the Synthesis of Product Knowledge Across the Lifecycle," V012T13A071. ASME, 2013. doi:10.1115/IMECE2013-65220.

[89] World Wide Web Consortium. OWL 2 Web Ontology Language Document Overview. 2009. Available from: http://www.w3.org/TR/2009/REC-owl2-overview-20091027/.

[90] Zentner, J., Ender, T., Ballestrini-Robinso, S., On Modeling and Simulation Methods for Capturing Emergent Behaviors for Systems-of-Systems, 12th Annual Systems Engineering Conference, October, 2009.

[91] Zimmerman, P., Model-Based Systems Engineering (MBSE) in Government: Leveraging the 'M' for DoD Acquisition, 2014 INCOSE MBSE Workshop January 25, 2014.

[92] zur Muehlen, M., D. Hamilton, R. Peak, Integration of M&S (Modeling and Simulation), Software Design and DoDAF, SERC-2012-TR-024, 2012.