Transforming System Engineering through Model-Centric Engineering

Technical Report SERC-2016-TR-109

September 1, 2016

Principal Investigator
Dr. Mark Blackburn, Stevens Institute of Technology

Research Team
Dr. Mary Bone, Stevens Institute of Technology

Sponsor: The Naval Air Systems Command,
The Deputy Assistant Secretary of Defense for Systems Engineering
Table of Contents

1 Introduction .......................................................................................................................... 3
  1.1 Objectives ......................................................................................................................... 7
  1.2 Scope ................................................................................................................................. 7
  1.3 Organization of Document ................................................................................................. 8

2 In-Process Summary .............................................................................................................. 10
  2.1 Characterizing Problem and Vision .................................................................................. 10
  2.2 SE Transformation (SET) – Perspectives on Clarifying the Focus .................................... 16
  2.3 Goal-Driven Plan ............................................................................................................... 17
  2.4 Working Sessions and Sponsor-Supporting Events ............................................................ 19
  2.5 Expanded Scope Under RT-170 ....................................................................................... 20
    2.5.1 RT-170 Task 1: Mission Engineering and Analysis using MDAO Methods .................. 21
    2.5.2 RT-170 Task 2: Decision Framework related to Cross-Domain Integration ................. 22
    2.5.3 RT-170 Task 3: Methods for Integrated Digital/Collaboration Environment ............. 22
    2.5.4 RT-170 Task 4 - Update System Engineering Transformation Roadmap – Task 4 .... 23

3 Task 1 – Model Cross-Domain Integration with underlying Single Source of Technical Truth (SSTT) 24
  3.1 Information Model for a Single Source of Technical Truth .............................................. 24
  3.2 Requirement Ontology Status ......................................................................................... 26

4 Task 2 – Model Integrity – developing and accessing trust in model and simulation predictions 27

5 Task 3 – Modeling Methodologies ....................................................................................... 28
  5.1 Modeling Examples Overview ........................................................................................ 29
  5.2 Modeling Examples ......................................................................................................... 30
    5.2.1 Table of Contents ....................................................................................................... 31
    5.2.2 Process/Methods ....................................................................................................... 31
    5.2.3 Package Hierarchy for Structuring and Organizing Model Information ..................... 34
    5.2.4 Mission Level Models ............................................................................................... 35
    5.2.5 System level models .................................................................................................. 38
    5.2.6 Activity Diagram of Dave Cohen’s Framework Process ............................................. 39
  5.3 Views and Viewpoints ...................................................................................................... 40
  5.4 Multidisciplinary Design, Analysis and Optimization ...................................................... 41
    5.4.1 Historical Context from Industry Discussions on MDAO ........................................ 41
    5.4.2 MDAO-relevant Methods and Tools ........................................................................ 42
  5.5 Capability and Operational-Level Modeling Guidelines ................................................. 44
  5.6 NAVAIR Study Views ..................................................................................................... 45
  5.7 Modeling and Methods for Uncertainty Quantification ................................................... 46
    5.7.1 Dakota Sensitivity Analysis and Uncertainty Quantification (UQ) ............................... 47
    5.7.2 An Overview of Quantification of Margins and Uncertainty .................................... 49
  5.8 Modeling Methods for Risk .............................................................................................. 51
    5.8.1 Predictive Models for Risk ......................................................................................... 51
  5.9 Controlled Natural Language Requirements information .............................................. 52

6 Task 4 – Define System Engineering Transformation Roadmap .......................................... 53

7 Integrated Framework for Risk Identification and Management ...................................... 55
  7.1 Risk Context ..................................................................................................................... 55
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.1.1</td>
<td>Risk of Consequence from Model Centric Engineering Transformation</td>
<td>56</td>
</tr>
<tr>
<td>7.1.2</td>
<td>Future Root Causes</td>
<td>57</td>
</tr>
<tr>
<td>7.2</td>
<td>Scope of the Risk Framework</td>
<td>57</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Risk Framework Captures Knowledge</td>
<td>59</td>
</tr>
<tr>
<td>7.3</td>
<td>Model Validation and Simulation Qualification</td>
<td>60</td>
</tr>
<tr>
<td>7.4</td>
<td>Improving the Integrity of Models</td>
<td>61</td>
</tr>
<tr>
<td>7.5</td>
<td>Model-Centric Methods and Tools Enable Approaches to Safety and Airworthiness</td>
<td>62</td>
</tr>
<tr>
<td>7.5.1</td>
<td>MCE Methods to Generate Failure Modes and Effects Analysis (FMEA) models</td>
<td>63</td>
</tr>
<tr>
<td>7.5.2</td>
<td>Mission-level Model-based Safety Analysis</td>
<td>64</td>
</tr>
<tr>
<td>7.5.3</td>
<td>Risk Reliance on Model Correctness and Accuracy</td>
<td>65</td>
</tr>
<tr>
<td>8</td>
<td>Part II Summary</td>
<td>66</td>
</tr>
<tr>
<td>9</td>
<td>Acronyms and Abbreviation</td>
<td>68</td>
</tr>
<tr>
<td>10</td>
<td>Trademarks</td>
<td>71</td>
</tr>
<tr>
<td>11</td>
<td>References</td>
<td>73</td>
</tr>
</tbody>
</table>
Figures

Figure 1. SE Transformation “Roll out” Strategy ................................................................................. 2
Figure 2. SE Transformation Phase II .................................................................................................. 4
Figure 3. SE Transformation Research Areas (SERC) ........................................................................ 5
Figure 4. Proposed Framework for New Operational Paradigm for Acquisition and Design .............. 6
Figure 5. Challenges/Focus Areas associated with Proposed Framework ........................................... 8
Figure 6. Integrated Environment for Iterative Tradespace Analysis of Problem and Design Space ...... 11
Figure 7. Dynamic CONOPS Integrated with Mission Simulations ...................................................... 12
Figure 8. Multidisciplinary Design, Analysis and Optimization Supports Tradespace Analysis Across Disciplines .................................................................................................................. 13
Figure 9. Integrate Multiple Levels of System Models with Discipline-Specific Designs .................... 14
Figure 10. Appropriate Methods Needed Across Domains .................................................................. 15
Figure 11. Need for Obtaining Digital Information Across the Domains ........................................... 16
Figure 12. Conceptual POAM Related to ISEE, SSTT, and MDAO .................................................... 19
Figure 13. Traceability and Scope of Data Collection of MCE Relevant Topics ................................. 21
Figure 14. Integrated Data Objects Partial Entity Relational Diagram .................................................. 25
Figure 15. Association to Requirements .............................................................................................. 26
Figure 16. Ontology and Requirement Manager Engine Prototype .................................................... 27
Figure 17. Example MDAO Model ....................................................................................................... 30
Figure 18. Table of Contents to Models and Diagrams ....................................................................... 31
Figure 19. Pre-modeling Guidelines .................................................................................................... 32
Figure 20. Containment Structure ....................................................................................................... 33
Figure 21. Simple MBSE Activity Diagram with Link to MDAO ............................................................ 34
Figure 22. Model Organization ............................................................................................................ 35
Figure 23. High-Level Mission Use Case ............................................................................................. 36
Figure 24. Textual Element of the Use Case ....................................................................................... 37
Figure 25. Mission-level Activity Diagram with Swim Lane Partitions .................................................. 38
Figure 26. Generic UAV Use Case Diagram with Actors .................................................................... 39
Figure 27. State Machine Diagram of Top-Level UAV Operational States .......................................... 39
Figure 28. Draft Activity Diagram of SE Transformation Framework ............................................... 40
Figure 29. Viewpoint .......................................................................................................................... 40
Figure 30. MDAO Compared with Legacy Tradespace Analysis ............................................................ 42
Figure 31. Canonical Problem & MDO “Architecture” based on Design Structure Matrix .................. 43
Figure 32. OpenMDAO Conceptual Example ................................................................. 44
Figure 33. Mission Context for System Capability ....................................................... 46
Figure 34. Dakota Framework Integration Wraps User Application ............................. 48
Figure 35. Example for Understanding Margins and Uncertainty ............................. 48
Figure 36. Pulling Together Concept Associated with QMU ...................................... 51
Figure 37. Bayesian Model Derived from Airworthiness Factors ................................. 52
Figure 39. Example Failure Modes and Effects Analysis Metamodel ............................ 66
Tables

Table 1. 2016 RT-157 Plan Objectives, Actions and Milestones................................................................. 18
Acknowledgments

We wish to acknowledge the great support of the NAVAIR and SERC sponsors, 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 immediate NAVAIR team, Jaime Guerrero, David Meiser, Jason Thomas, Chris Owen, Jeff Smallwood, Michael Gaydar and James Light who has worked closely on a weekly basis in helping to collaboratively research this effort.

We also want to thank all, currently more than 220 stakeholders that participated in over 30 organizational discussion and 24 working session, and many follow-up sessions supporting the new transformation. There are so many contributor, supporters and direct stakeholders that supported this effort, we wish to recognize them all. Please see our prior report for earlier contributors. We sincerely apologize if we have missed anyone else that has supported our efforts.

Bill Brickner  Eric (Tre’) Johnsen  John McKeown  Philomena Zimmerman
Brandi Gertsner  Fran Chamberlain  John Quartuccio  Richard Yates
Brian Nolan  Gary Strauss  Keith Carter  Ron Carlson
Brent Gordon  Gary Witus  Lance Hernandez  Scott Lucero
David Fields  Geethesh Kukkala  Megan Clifford  Shahram Bavani
Dennis Reed  Jae Pfeffer  Nathaniel Barkley  Stu Young
Dinesh Verma  James Carrol  Tom Blakely
Executive Summary

This is the interim report of the Systems Engineering Research Center (SERC) research task RT-157. This research task (RT) is addressing the next phase of research challenges following the prior efforts under RT-48/118/141 that informed us that model-centric engineering (MCE) is in use and adoption seems to be accelerating. Model-centric engineering\(^1\) can be characterized as an overarching digital engineering approach that integrates different model types with simulations, surrogates, systems and components at different levels of abstraction and fidelity across disciplines throughout the lifecycle. Industry is trending towards more integration of computational capabilities, models, software, hardware, platforms, and humans-in-the-loop. The integrated perspectives provide cross-domain views for rapid system level analysis allowing engineers from various disciplines using dynamic models and surrogates to support continuous and often virtual verification and validation for tradespace decisions in the face of changing mission needs.

NAVAIR senior leadership confirmed in late 2015 that the research findings and analysis have validated their vision hypothesis stated at the System Engineering Transformation kickoff meeting of RT-48. They concluded that NAVAIR must move quickly to keep pace with the other organizations that have adopted MCE and who continue to evolve at an accelerating pace enabled by the advances in computational and modeling technologies, and improved methods. NAVAIR must also transform in order to continue to perform effective oversight of weapon system development by primes that are using modern modeling methods for system development. The risks of not moving forward include making acquisition decisions with progressively less technical-truth insight and the proliferation of disparate, redundant and stove-piped data and models, and lacking MCE capabilities and knowledge needed to understand an increasingly complex problem and design space.

The path forward has challenges but also many opportunities, both technical and sociotechnical. It must include a modeling framework with high performance computing (HPC) that enables single source of technical truth (SSTT), integration of multi-domain and multi-physics models, and provides for a method for model integrity. The modeling and infrastructure for a digital engineering environment is a critical step to enable a SSTT. While there are literally thousands of tools, they are often federated and there is no one single solution that can be purchased. Every organization providing inputs to this research has had to architect and engineer their model-centric engineering environment. Most have selected commercial tools and have developed the integrating fabric between the different tools, models, and data. This approach often uniquely positions them with some advantages among the rest. Some organizations have encoded historical knowledge in reference models, model patterns to embed methodological guidance to support continuous orchestration of analysis through new modeling metrics, and automated workflow.

The kickoff of RT-157 in January 2016 defined a research plan to investigate challenge areas include but are not limited to:

- Cross-domain integration of models to address the heterogeneity of the various tools and environments
- Model integrity to ensure trust in the model predictions by understanding and quantifying margins and uncertainty

---

\(^1\) DASD has increased the emphasis on using the term Digital Engineering. A draft definition provided by the Defense Acquisition University (DAU) for DE is: **An integrated digital approach that uses authoritative sources of systems' data and models as a continuum across disciplines to support lifecycle activities from concept through disposal.** This definition is similar to working definition used throughout our prior research task RT-48/118/141 for Model Centric Engineering (MCE).
Modeling methodologies that can embed demonstrated best practices and provide computational technologies for real-time training within digital engineering environments

Multidisciplinary System Engineering transformation roadmap that looks across:
- Technologies and their evolution
- How people interact through digitally enabled technologies and new needed competencies
- How methodologies enabled by technologies change and subsume processes
- How acquisition organizations and industry operate in a digital engineering environment throughout the phases of the lifecycle (including operations and sustainment)
- Governance within this new digital and continually adapting environment

In March of 2016, there was a Change of Command at AIR 4.0 (Research and Engineering). NAVAIR decided to accelerate the Systems Engineering (SE) transformation. NAVAIR has created a broader Plan Objectives, Action and Milestones (POA&M) for 2016-2018 exceeding 600 items. Some of those items correspond to RT-157 tasks. NAVAIR would like more SERC involvement to assist in the SET Acceleration and proposed new tasking for a follow-up RT-170. Notionally as shown in Figure 1, the broader POA&M has a layered approach where the needed research provides analyses into NAVAIR enterprise capability, but builds on efforts for cross-domain model integration and model integrity (per RT-157). While the SERC research was directed to focus on the Program of Record (POR)/systems level, a NAVAIR strategy for accelerating capability delivery to the warfighter is also looking at a new operational paradigm for conducting acquisition and design. This report discusses the accomplishments to date and alignment of the future objectives to the new and evolving NAVAIR strategy based on MCE/DE.

![Figure 1. SE Transformation “Roll out” Strategy](image-url)
1 INTRODUCTION

In 2013, Naval Air Systems Command (NAVAIR) at the Naval Air Station, Patuxent River, Maryland initiated research into a Vision held by NAVAIR’s leadership to assess the technical feasibility of a radical transformation through a more holistic model-centric system engineering (MCSE) 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 included the evaluation of emerging system design through computer (i.e., digital) models.

Through Systems Engineering Research Center (SERC) research tasks (RT-48, 118, 141) there was considerable emphasis on understanding the state-of-the-art through discussions with industry, government and academia [19] [29]. The team comprised of both NAVAIR and SERC researchers conducted over 30 discussions, including 21 on site, as well as several follow-up discussions on some of the identified challenge areas and approaches for a new operational paradigm between government and industry.

In 2015, the NAVAIR leadership concluded that they must move quickly to keep pace with the other organizations that have adopted MCE as the pace of evolution is accelerating by the enabling technologies. NAVAIR made the decision to press forward with a Systems Engineering (SE) Transformation. That effort was started in January of 2016 under RT-157 had four tasks as shown in Figure 2:

- Task 1 – Model Cross-Domain Integration with underlying Single Source of Technical Truth (SSTT)
- Task 2 – Model Integrity – developing and accessing trust in model and simulation predictions
- Task 3 – Modeling Methodologies aligning with the roll out of technologies defined under Task 4
- Task 4 – Define System Engineering Transformation Roadmap
In March of 2016, there was a Change of Command at AIR 4.0 (Research and Engineering). NAVAIR decided to accelerate the SE transformation (SET). NAVAIR has created a broader POA&M, now exceeding 600 items. Some of those items correspond to RT-157 task. NAVAIR has plans for more SERC involvement to assist in the SET Acceleration. Notionally as shown in Figure 1, the broader POA&M has a layered approach where the needed research provides analyses into NAVAIR enterprise capability, but builds on efforts for cross-domain model integration and model integrity (per RT-157). While the SERC research was directed to focus on the Program of Record (POR)/systems level, a new NAVAIR strategy for accelerating capability delivery to the warfighter is looking to better assess the value and risks of system and system of systems (SoS) capabilities, potentially distributed across platforms to mission and campaign needs in a more dynamically changing environment. Therefore, NAVAIR believes the following areas are candidates for SERC research and characterized in RT-170; these layer on top of the other dimensions of the research as shown in Figure 3:

- Prioritization and Trade-off Analysis
- Concept Engineering
- Architecture & Design Analysis
- Design & Test Reuse and Synthesis
- Active System Characterization
- Human-System Integration
Finally, our sponsor Dave Cohen proposed a new operational framework, which is shown in Figure 4. The framework is being assessed and refined in order to support a new operational paradigm to mission engineering, analysis and acquisition, which would be led by NAVAIR with a collaborative design effort led by industry. Briefly the concept under consideration for transforming from a document-centric process with monolithic reviews to an event-driven model-centric approach involves:

- **Steps 1-4** represent a concept for collaborative involvement between Government and Industry to assess mission and System of Systems (SoS) capability analyses
  - Involve industry in SoS capabilities assessments during mission-level analysis (to the degree possible)
  - Iteratively perform tradespace analyses of the mission capabilities using approaches such as Multidisciplinary Design, Analysis and Optimization (MDAO) as a means to develop and verify a model-based specification
  - Synthesize an engineering concept system model characterized as a model-centric specification and associated contractual mechanism based on models or associated formalism
- **Steps 5-7** (there is some uncertainty about the iterative set of steps)
  - At step 5, industry will lead a process to satisfy conceptual model addressing the Key System Attributes (KSAs)², with particular focus on Performance, Availability, Affordability, and Airworthiness to create and Initial Balanced Optimized Design

² We have been informed that Key System Attributes are being substituted for Key Performance Parameters in the Joint Capabilities Integration and Development System (JCIDS).
Industry too applies MDAO and there is a potential need to iterate back to Steps 1-4 if tradespace analyses of the solution/system for the program of record (POR) could not achieve mission-level objectives.

All requirements are tradeable if they don’t add value to the mission-level KSAs.

These are asynchronous activities in creating an Initial Balanced Optimized Design.

Government and Industry must work together to assess “digital evidence” and “production feasibility”:

- How does/can government participate in these continual event-driven and objective evaluation steps?
- How to judge evolving maturity of design?
- How can government interact to provide value without impendence?

This report covers both the research performed against the RT-157 objectives, and the request of our sponsor that has expanded to address the needs of the SE transformation, specifically in the context of the new framework (Figure 4).
1.1 Objectives

As shown in Figure 3, the scope of these research task areas has expanded. The specific tasks are being prioritized using the NAVAIR POA&M and the new proposed framework for how NAVAIR will operate with industry using MCE (part of the SE Transformation Roadmap). We are supporting the research using a case study based on a conceptual Unmanned Air Vehicle (UAV). This case study serves as a surrogate for the pilot project identified in Figure 1. This example directly supports some of the needs for Task 3 (Modeling Methods) and Task 4.

We are using publically available information such as: Conceptual Design and Aerodynamic Analyses of a Generic UCAV Configuration [81] to construct examples and reference models, including SysML, MDAO system examples, and mission-level CONOP scenarios such as:

- Surveillance
- Refueling
- On-board UAV refueling

This approach supports a research objective to inform competencies (Task 4) with reference models and modeling methods. The examples can be “reference models” of modeling patterns or surrogate. This case study supports the three cross-cutting critical items from RT-157, but extends it to the mission/SoS level:

- Cross-domain and multi-physics model integration, and the associated methodologies
- Technologies to establish and quantify model integrity
- High Performance Computing (HPC), which enables the previous two bullet points

As reflected in Figure 4 there are four elements of research that relate to the critical enablers that extend the RT-157 task as defined in the RT-170 proposal that has been submitted, but not yet awarded, including:

- Mission Engineering and Analysis using MDAO methods and applicable operational, capability, and system models
- Decision framework related to cross-domain integration through the single source of technical truth (SSTT)
  - Provides a basis for an objective approach to assess design maturity based on an ontological representation of the system using open semantic web technologies
- Integrated digital/collaboration environment capabilities and operational models both within NAVAIR and industry
  - This includes the development of methods and reference models to enrich workforce understanding of MCE methods, models and tools
- Update the SE Transformation Roadmap
  - This should specifically address the need for new workforce skills
  - Needs to address iterations (currently three planned) as reflected in Figure 1

1.2 Scope

Given the objectives, there are new specific directives by the sponsor to assess the new framework and identify challenges and focus areas to develop a prioritization for the research. The current incomplete version of the challenge and focus areas, which is being developed jointly by the SERC and NAVAIR teams is shown in Figure 5. This has been shared with DASD (SE) and the SERC leadership. At least two key objectives have been prioritized and assigned to the SERC team that will govern some of the focus for the remainder of RT-157:
- Can requirements be represented as model?
  - And if so how – can we use the case study to help show, in the various ways, models can be used to support requirements, constraints, validation, and verification planning?
  - Can modeling help understand completeness and consistency of requirements?

We believe there are several other questions and are modeling the framework to support our analysis. NOTE: we know the content in Figure 5 is not readable, but the content has not yet been completed nor approved. The items in red are being removed/replaced, and we are considering a different format. Please contact us if you would like to get a version of the challenges.

### Figure 5. Challenges/Focus Areas associated with Proposed Framework

#### 1.3 ORGANIZATION OF DOCUMENT

Section 1 provides an overview of the context for the needed research, objectives, expanded scope and organization of this report.

Section 2 provides the in-process summary of our efforts, findings, analysis and recommendations including key aspects from RT-48/118/141. This sections also briefly summarizes the expanded scope of our research under the newly awarded RT-170.

Section 3 describes approach and results of developing information models underlying the single source of truth and a requirement ontology and requirement manager prototype.

Section 4 describes the need for and approaches to Model Integrity – developing and accessing trust in model and simulation predictions.
Section 5 describes the modeling methodologies, including examples and demonstrations created to illustrate mission, system, enterprise and reference models, including example and methods for Multidisciplinary Design, Analysis and Optimization.

Section 6 discusses the roadmap.

Section 7 provides some cross-cutting information on risk that relates to model integrity and modeling methods.

Section 8 provides conclusions with a brief summary of the planned next steps.
2 IN-PROCESS SUMMARY

This section provides some context into the concerns of the NAVAIR leadership in moving forward with a SE transformation. This section also provides a summary of the in-process information required as an intermediate deliverable for RT-157. Similar to the approach used in RT-48/118/141, this section provides a high-level summary of the plans, results and deliverables. Part II of this report provides additional details on each task.

2.1 CHARACTERIZING PROBLEM AND VISION

The RT-141 final report [19] generalized capabilities heard by many organizations [6] [7] [8] [36] [48] [68] [74] [105] and characterizes a canonical reference architecture of an Integrated MCE Environment, as shown in Figure 6. The following provides some perspectives and capabilities of this vision concept:

- Provides appropriate views for the various stakeholder
- Stakeholders have views into the Single Source of Technical Truth (SSTT)
  - Using rich modeling interfaces for those with expertise in modeling
  - Using rich “web” interface, which today provides support for graphics, integrated with structure inputs, generated textual views and 3D model viewing [109]
- MDAO layer provides for problem and design space exploration of
  o Physics-based models
  o Integrity-based models
  o Cost and scheduling models
  o Risk models
  o Various “illities” models
  o Including surrogates and components
- Enabled by High Performance Computing (HPC)
- Semantically rich linkages between data and information in the SSTT provides for continuous workflow orchestration – enabled by HPC
- Document generation is enabled by
  o Semantically rich links to information in the SSTT
  o Templates that formalize patterns for requirements, contracts, etc.
- Enabling technologies such as machine learning provides a virtual knowledge librarian that assist users guided by embedding knowledge and training
- Contractor and collaborators have a secure means to plugin to view or share digital information as a new paradigm for interactions
- This view of the Designing System provides links downstream to fully link Product Lifecycle Management (PLM)
Our working sessions, as summarized in Section 2.4, continue to initiate discussions about what the SE transformation is and its implications moving forward, benefits and challenges. We have used the following scenarios to supplement our characterization of a future vision state shown in Figure 6. It has helped provide additional perspectives moving from a mission-perspective to several systems-perspectives down to specific design disciplines. It also provides a means to identify some of the challenges associated with the four tasks of RT-157, but also other needs defined in RT-170.

As shown in Figure 7, we have seen the use of the Net-Centric Evaluation Capability Module (NECM) that uses modeling and simulation and a Study Views method to structure the development of needed mission capabilities. These capabilities support some aspects of the Joint Capabilities Integration and Development System (JCIDS) to analyze joint mission threads in near real-time and automate net-ready Key Performance Parameter (KPP) analysis. Currently this information (and others) are used as input to develop DoDAF model, which are focused primarily on the net-ready capabilities. However, there are more concerns in developing a weapon system than just the net-ready capabilities. Therefore, if we want to support the vision of the NAVAIR sponsor to have a deeper system and component-level analysis as it relates to the value to key KSA/KPPs relative to the mission, we need better integration to high-level fidelity models at the system levels as described in the scenarios below. We believe there are opportunities and benefits to better link the NECM capabilities into dynamic simulations of both mission and system capabilities to create more dynamic operational representations of the concepts of operation (CONOPS).

---

3 Image credit: AGI.
Those dynamic capabilities reflected in Figure 7 that are linked into the study views map to higher-level fidelity views related to specific design disciplines as shown in Figure 8. At this level, we want to improve the tradespace by doing MDAO. Most organization that develop aircraft systems have been using MDAO for over 10 years, more focused at the system level. Such capabilities allow for 1000x the number of design excursions as has been done with traditional approaches in the past. These types of MDAO approaches provide for some amount of cross-domain analysis, however, we want to develop more comprehensive approaches and be systematic about covering the tradespace at the mission and system levels, at least for the critical KSAs. All of the main contractors to NAVAIR use these capabilities, and we shared publically known information with attendees about such usage at our working sessions. MDAO can also be used at the mission level as reflected in Figure 4. There are opportunities in research for developing and improving MDAO methods [82].

4 Image credit: Phoenix Integration
Another area of opportunity is to improve the integration of architectural, system and component models across the domains, and better link with other modeling and simulation capabilities targeted to specific disciplines. These “architectural” models may be developed using DoDAB to characterize mission capabilities and operational views. At the system level they may be developed using Model Based System Engineering (MBSE) methods and be represented in standard modeling languages such as SysML [100]. The linkages between the MBSE and design disciplines is often not precisely represented, with a few exceptions. When it is done using tool-to-tool integration, such integrations can be rather susceptible to tools updates [32]. We believe there are opportunities to address this need in more tool agnostic ways using semantic web technologies; this is one area of research supporting cross-domain model integration (Task 1).
The key reason for the need for cross-domain model integration (Task 1) is the underlying complexity needed to accomplish the scenarios associated with Figure 8 and Figure 9. In addition, our research as illustrated by the DARPA META project [7] has shown that methods are needed to ensure that the tools provide the expected automation, efficiencies, and produce the desired information. This points to the need for both methods (Task 3) and because many of the modeling and simulation capabilities that may be integrated into an MDAO workflow can be modeling and simulation capabilities, they require some type of assessment to ensure the integrity of the predictions (Task 2).

We believe there are research challenges to better quantify design margins, parameter uncertainties, and system performance sensitivities associated with physics-based digital models. There are opportunities and challenges in integration of relevant multi-physics modeling and simulation, need for earlier high-fidelity models, and means to assess reduced-order models. In addition, there are needs for determining optimal risk/cost tradeoff for continual Verification, Validation and Accreditation (VV&A) or alternative means for assessing trust in model and simulation predictions.

As shown in Figure 10 [42], there can be a very large set of tools that can be used to develop the needed data and information across all of the domains⁵. Therefore, it is important that appropriate methods are applied to the selected tools that are assembled for use on a project or program. As a secondary objective that is being demonstrated as leading edge approach by NASA/JPL is to ensure models are created that comply with established modeling patterns; their approach transforms the model information into a tool-neutral SSTT based on ontologies, and then uses standard semantic web technologies to apply checks to ensure completeness and consistency [74]. Some of our deliverables are providing the building blocks to

---

⁵ For example, in an inventory analysis of modeling and simulation tools used at NAVAIR, there was more than 200, and we were told it was incomplete.
accomplish these types of checks and are planned to be integrated into the NAVAIR Integrated Systems Engineering Environment (ISEE) as discussed below.

Cross-domain methodologies ensure tool usage produces complete and consistent information compliant with ontologies of SSTT

Another challenge underlying the reason for the need for cross-domain model integration (Task 1), as rendered notionally in Figure 11, is that the current competencies may support one or more domains may often “buy all the data,” often referred to as Contract Data Requirements List (CDRLs); this practice is aimed at reducing the potential future risks. Currently, there are limited ways of understanding the value of that data to the mission or the implication of design decisions across the domains. Models, modeling, and computationally enabled concepts such as the use of precise models that are linked across domains in the SSTT should provide a means to begin to understand the value, risk and uncertainty of information relative to the mission. This is a specific objective articulated by the sponsor. However, this also leads to another new need in how digital engineering impacts the language and information that is put on contract in a statement of work (SOW) or requests for proposal (RFP). Therefore, we too are looking at potential approaches this need as that relates to the information that flows between step 4 and 5 in Figure 4. We have an effort to look at approaches to formalize contractual language, and may need to define how a proposal response will be evaluated using modeling and digital engineering information.
2.2 SE TRANSFORMATION (SET) – PERSPECTIVES ON CLARIFYING THE FOCUS

We have also been requested to contribute to an effort to create an executive level message that can be used by senior leadership to describe the SE Transformation (SET). Like most complex enterprise, mission and system problems there are many stakeholders and they all have different concerns and views on the solution and/or vision. We share some perspectives from some of the leaders of the SE transformation:

- Senior Executive for Research and Engineering - Emphasizes Digitalization and Virtualization
  - He also discusses it in terms of Better, Faster, Cheaper
- Dave Cohen - Emphasizes the ability for models to represent a higher level of abstraction of the system including the multi-physics aspects in order to get to an Integrated Test Vehicle (per SET Framework)
  - We also know based on the framework that Dave is concerned with continuous collaboration between Government and Industry and event-driven decision making
- Jaime Guerrero - Emphasizes the underlying "data" (Information Model) of the integrated information that represents "all" aspects of the system, mission, users and environment in the Single Source of Technical Truth (SSTT)
  - Jaime also realizes that this perspective is about Systems Engineering, including principles and methods carried out in terms of more precise models and standard languages
  - Integrated views across all of the domains including risk, uncertainty, and new metrics for understanding “design maturity” (recognizing that some and maybe most competencies still like to think in their stove pipes)
  - Leveraging computational capabilities to let the computer do work that in a document-centric world is done primarily by humans
The digitalization in terms of precise models allows computers to help find defects to make the system “Better.” Computers also provide the means to analyze 1000x number of trades, again based on precise models and the associated simulations leading to a “Better” design, both in terms of multi-physics, but also in terms of other cost and “ilities” models.

The digitalization in terms of precise models also allows computers to do work that has in the past been done by humans allowing work to be done “Faster.”

The precise digitalization including the increased emphasis on representing the cross-domain relationships and dependencies associated with the competencies and disciplines provides for a SSTT where work is event-driven (“Faster”), allowing for the continuous Virtualization of meetings (“Faster”) to eliminate the monolithic and costly reviews (“Cheaper”).

As a community, perhaps we should create a Systems Engineering Transformation Manifesto (e.g., Manifesto for Agile Software Development is quite well known). A manifesto is a written statement where one (or a group) publicly declare their:

- Intentions (what you/we intend to do)
  - Change the operational paradigm for acquisition of system and system of systems
- Opinions (what you/we believe; stance on a particular topic)
  - We can operate in a more collaborative paradigm
  - Computationally enabled Systems Engineer allows us to deal with complexities not possible through traditional document-centric processes (“Better”)
  - Process of models provides for early Validation of Requirements (“Better”)
  - Resulting models produces Verification threads to support Verification planning that can serve as a basis of estimate earlier (“Better”)
  - Models are reusable from program-to-program (“Faster, Cheaper”)
- Vision (the type of world that you dream about and wish to create)
  - See Figure 6 and associated characterization

There are many organizations in DoD that might benefit from coming together with a unified vision on Systems Engineering Transformations.

### 2.3 GOAL-DRIVEN PLAN

RT-157 started in February using a goal-oriented method to develop a Plan Objectives, Action and Milestones (POA&M) for the 2016 research, with a mapping to the roadmap categories: (T) Technologies, (M) Methods, (C) Competencies (see Table 1). This plan was developed before the broader POA&M that was discussed in Section 1. This plan was based on the goal to establish a rigorous foundation for a semantically precise and tool agnostic framework for the SSTT (Task 1). These detailed tasks are contributing to the extension of the Integrated System Engineering Environment (ISEE) as shown in Figure 12. There was also the assumption that there would be an evolutionary adoption of modeling tools and methods at NAVAIR, and the focus would start with improved formalization of requirements that would trace to models, risks, and evidence. Note also that there is a roadmap tag mapping to technology (T), methods (M), competencies (C). We had no specific tasks that was to address interactions with contracting organizations or a new approach to governance. However, the new framework does bring in plans for a new operational model between government and industry. In addition, the change of command altered this plan implies some changes in governance as reflected by the framework (Figure 4).
### Table 1. 2016 RT-157 Plan Objectives, Actions and Milestones

<table>
<thead>
<tr>
<th>ID</th>
<th>What</th>
<th>Why</th>
<th>Roadmap Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>See Goals Tab</td>
<td>List general goals, priorities and dependencies</td>
<td>T,M,C</td>
</tr>
<tr>
<td>1</td>
<td>Requirements Engineering Method</td>
<td>There are multiple capabilities involved in requirements engineering, including ontology-driven requirement engineering to ensure consistency and completeness of information captured; another goal that might take longer to implement is to embed the methodological guidance in the Requirement manager implementation</td>
<td>M</td>
</tr>
<tr>
<td>1a</td>
<td>Requirements Ontology</td>
<td>Characterizes in a tool agnostic way the information that must be captured, in addition to representing the attributes of consistency and completeness</td>
<td>T</td>
</tr>
<tr>
<td>1b</td>
<td>Requirements Manager Tool Trade</td>
<td>Determine the tools that can perform multiple activities of requirements engineering from elicitation, traceability, through version control, role-based access, security</td>
<td>T, M</td>
</tr>
<tr>
<td>1c</td>
<td>Requirement Manager</td>
<td>Implementation and integration into the SSE and single source of technical truth</td>
<td>T, M</td>
</tr>
<tr>
<td>2</td>
<td>Risk Method</td>
<td>The risk method will leverage the formality defined in a risk ontology that integrates both with requirements and evidence to support a rigor risk approach that is based on evidence of work, including the incorporation of evidence provided as models</td>
<td>M</td>
</tr>
<tr>
<td>2a</td>
<td>Risk Ontology</td>
<td>The key classes from the existing process are already defined, but will be formalized into an ontology that links to both requirements and evidence that will be stored in the SSTT</td>
<td>T</td>
</tr>
<tr>
<td>2b</td>
<td>Evidence Ontology</td>
<td>This is information that is related to the information from the checklist; this will be formalized to link to both requirements and risk. This will formalize the risk process(e.g., low evidence implies high risk)</td>
<td>T</td>
</tr>
<tr>
<td>2c</td>
<td>Requirements to Risk Mapping</td>
<td>Relate risk to requirements</td>
<td>T</td>
</tr>
<tr>
<td>2d</td>
<td>Risk Manager</td>
<td>Implementation and integration of risk viewpoints and functions into the SSTT</td>
<td>T, M</td>
</tr>
<tr>
<td>2e</td>
<td>Requirements to Evidence Mapping</td>
<td>Relate evidence to requirements</td>
<td>T</td>
</tr>
<tr>
<td>2f</td>
<td>Risk to Evidence Mapping</td>
<td>Relate risk to evidence</td>
<td>T</td>
</tr>
<tr>
<td>3</td>
<td>Knowledge</td>
<td>This involves the integration of embedded training, examples, modeling patterns, reference models, tradespace analyses; engineers work well with examples. There is currently not much available, and implementation of this concept needs to attempt to embed the methodologies into different tools, as is the currently the case with SETR manager</td>
<td>T, M, C</td>
</tr>
<tr>
<td>4</td>
<td>Multidisciplinary Design, Analysis and Optimization</td>
<td>MDAO is a systematic approach to tradespace analysis. This task in 2016 involves informing people about the concept, methods and tools options.</td>
<td>C, M</td>
</tr>
</tbody>
</table>
2.4 Working Sessions and Sponsor-Supporting Events

A component of the research and required deliverables are conducting working sessions that inform the NAVAIR team about progress against the plan. These working sessions also inform the team about relevant information and feedback to scope the deliverables in the context appropriate for NAVAIR; this has been especially important given the recent changes under SE transformation. We also use bi-weekly drumbeats to share status and updates. Each working session has a defined agenda, and detailed meeting notes are provided to our sponsor. The following provides a summary of the working sessions and other events, and a brief description of the contributions to the tasks and deliverables.

- Working session #18: 2/4/2016
  - Developed the goal-driven prioritization plan for RT-157 and confirmed the priorities with the NAVAIR team and sponsors
  - Discussed the concept for developing the ontology underlying the requirement manager (top-level priority)

- Working session #19: 3/3/2016
  - Presented a session on methods for Multidisciplinary Design, Analysis and Optimization methods and tools
  - Presented evidence and example for a concept based on Controlled Natural Language Requirements that will likely be necessary as a supplement for models

- Working session #20: 4/7/2016
  - Discussed the new plan of the SE Transformation acceleration
  - Discussed POA&M for the RT-157-specific discussed in Section 2.3
  - Discussed the importance of modeling methods before tool selection
Discussed use of reference models as curated knowledge that may alter the way training is conducted, and must be considered in the creation of the ISEE

Discussed the approach for developing the underlying Information Model for the SSTT

Presented a working demonstration of the requirement ontology and requirement manager prototype and provided software prototype and ontology to NAVAIR (a bonus deliverable)


Discussed concepts and implications associated with the SE transformation and framework (see Figure 4)

Working session# 22: 6/7/2016

Discussed first version of the new framework and implications of developing a POA&M for planning next few years of the SE transformation that included all of NAVAIR and pilot projects as shown in Figure 1

Initiated the discussion for follow-on research task now in RT-170, which was awarded on 1-Sep-2016

Working session #23: 7/14/2016

Discussed underlying meaning of the SE transformation (see Section 2.2)

Discussed of framework needs and challenges

• Implications on specification and contracting

Presented conceptual UAV as example case study with models for presenting modeling methods

Change in needs such as modeling examples

Change in collaborator from Wayne State to University of Maryland

Working session #24: 8/8/2016

Presented modeling method examples and demonstrations

• SysML at mission, system, and enterprise

• Reference models

• MDAO example of a UAV performance workflow

• MDAO webinars discussing usages by industry confirming use of MDAO by organization that contract to NAVAIR


Dave Cohen presented information that was pre-framework


SERC Executive Advisory Board: 6/29/2016

Presented results of NAVAIR research results and impacts

Dave Cohen was able to present early version of his framework

2.5 EXPANDED SCOPE UNDER RT-170

Although RT-170 was only recently awarded, it is acknowledged that there are many possible hurdles beyond technical feasibility (e.g., organizational adoption, training, usability, etc.), and we have had to adjust our plans and work to align with the new priorities due to the accelerated SE transformation. The path forward includes adjustments to the roadmap to factor in some of these other considerations. The concept proposed by the framework (Figure 4) has changed some of our assumptions about the SSTT.

The actual statement of work identified research needs relating to the cross-cutting concerns of the lifecycle and modeling environment and infrastructure such as:
- Prioritization & Tradeoff Analysis
- Concept Engineering
- Architecture & Design Analysis
- Design & Test Reuse & Synthesis
- Active System Characterization
- Human-System Integration
- Agile Process Engineering (new)

As shown in Figure 13 (columns to the right), these lifecycle perspectives were in the RT-141 final technical report, and as shown by the traceability, these needs cross many MCE relevant topics. All of these may be reasonable research needs, but we aligned the proposed task with the available level of funding to the key needs defined in the framework (Figure 4). These align the proposed task with our understanding of the sponsors priorities. We briefly summarize the new proposed RT-170 tasks.

![Figure 13. Traceability and Scope of Data Collection of MCE Relevant Topics](#)

### 2.5.1 RT-170 Task 1: Mission Engineering and Analysis using MDAO Methods

This task investigates factors relating to the relative value and priority of high-level capabilities of the system under development (some of which might be assessed under RT 170 Task 2). It investigates dynamic representations of mission and campaign analysis and defines methods for mapping to MCE-relevant capability representations in contrast to the traditional Capability Development Document (CDD). This should include modeling different viewpoints for capability views, operational views that map to system views.

In the context of the proposed framework shown in Figure 4, the concept of MDAO are being consider from at least three points of view, such as:

- Support validation of the model-based specification
  - Ensure completeness by tracing from the workflows to the capability threads
- Ensure an understanding of the boundary conditions against the KSAs to bound, quantify risk and surface potential anomalies
- Provide a means to look at the constraints imposed across domains
- Provides means for simulating dynamic behaviors, spanning multiple engineering domains to be able to balance the tradeoffs of performance, availability, affordability and airworthiness across domains
- Perform sensitivity analyses to assess the value or risk of different capabilities across domain/disciplines
- Support data-driven decisions with engineering technical data and information that is produced, not just documents (supporting RT-170 Task 2)
  - Capture and organize prior analysis of tradespace
  - Support reuse of data and analysis to perform regressions for the inevitable situations of evolving specifications
  - Trace to capability threads associated with specification
  - Provide means to integrate different resources (e.g., simulations, surrogates) from different sources (government and/or contractor)

### 2.5.2 RT-170 Task 2: Decision Framework Related to Cross-Domain Integration

The SSTT provides a basis for an objective approach to assess design maturity based on an ontological representation of the system using standards-based semantic web technologies. This will provide the means for assessing completeness and consistency across different models, developed using different languages and methodologies (as reflected in Task 3). This task will leverage semantic web technologies for creating an information model to demonstrate concepts for reasoning about conceptual models and design model maturity, which is tool neutral.

This task will extend the work with the requirements ontology and information model derived from the CORE model of Tier 3 artifacts developed under RT-141/157.

### 2.5.3 RT-170 Task 3: Methods for Integrated Digital/Collaboration Environment

This task focuses on the methodology transition from document-centric to model-centric in part to enhance our understanding/analysis capability of the increasing complexity in tactical systems. This specifically relates to, but is not limited to the methods used to support RT-170 Task 1 and RT-170 Task 2. We are also interested in model-based alternatives to specification representation and the ability to “generate requirements” that would lead to a digital representation of contractor input leading into Step 5 of the framework. This includes but is not limited to:

- MDAO workflows
- Model Based System Engineering (MBSE): Operational, Capability, Systems views
- Model Based Engineering (MBE): Discipline-specific, mechanical, electrical, controls, etc.
- Model Based “ilities” (MBX): Fault-trees, Bayesian, etc.
- Risk/Cost models

Finally, we also want to plan for the use and development of model patterns, model references within their environment to embedded knowledge, and methodological guidance to support continuous orchestration of analysis through modeling metrics, and automated workflow into the integrated environment. The case study should produce example models, methods, and reference models to enrich workforce understanding of MCE methods, models and tools. These efforts should support the evolution
and experimentation with the Integrated System Engineering Environment (ISEE), and define goals and requirements for the ISEE.

### 2.5.4 RT-170 Task 4 - Update System Engineering Transformation Roadmap – Task 4

The RT-157 roadmap task will be continued under RT-170.
Part II: Task Detail Summary

The material in Part II provides a summary of some of the task details, including information shared during some of the working sessions. An extensive amount of material covered in Part II of the RT-141 final report [19] still provides relevant information to this research, but has not been integrated into this interim report.

3 TASK 1 – MODEL CROSS-DOMAIN INTEGRATION WITH UNDERLYING SINGLE SOURCE OF TECHNICAL TRUTH (SSTT)

As discussed in Section 2, understanding the impacts related to cross-domain integration is a key need and challenge; these concerns can impact decisions made by different disciplines and competencies, and can be a critical risk, especially as systems increase in complexity. Traditionally the cross-domain implications surface during integration and test, which is typically late in the life cycle, and where changes can be costly. Today, this is an acknowledged problem and challenge. The solutions are often believed to be better standards for tool integration, but as discussed earlier tools continually change and the integrations become brittle [32]. Newer approaches based on data interoperability as a means of sharing information using standards and tool neutral approaches are emerging as being a better approach, and this is the approach we are pursuing.

3.1 INFORMATION MODEL FOR A SINGLE SOURCE OF TECHNICAL TRUTH

The concept for developing a SSTT is directly related to identifying the NAVAIR relevant information within the domains of the competencies and their relationships within and across the domains. We selected information from several sources in RT-141 final report to explain the evolving approach used by NASA/JPL. Crain [41] provided a process to explain how to approach the problem of understanding the underlying “data” for the producing systems. Start by identifying the objects (classes in an ontology), object properties, and object relationships. Figure 14 provides a perspective on some of the “data objects” and their associated relationships that are relevant to the enterprise at NASA/JSC; similar objects are relevant to NAVAIR too. We have identified about 300 “objects” that are needed to define a NAVAIR-relevant Information Model that underlies the SSTT. This information was collected by analyzing the artifacts collected as part of the Tier 3 products from the “As Is” effort of RT-48/118/141, and created in a CORE (Vitech) model.
We have discussed ontologies as a means of creating a tool-neutral information model. An ontology is a conceptualization for a domain with the associated relationships as shown in Figure 14. What is at least as important is that an ontology can be represented in the standard language OWL (Web Ontology Language, actually OWL2) where open and standard Semantic Web Technologies (tools) can be used to store, update, delete, query, and reason about consistency and completeness; such information is stored in a type of database called a triple store. An ontology can be thought about as a schema in the database for the data related to an ontology. In addition, we can relate different domain ontologies, which reflect on the cross-domain dependencies. This approach is what we call tool agnostic, but can map to any tool that stores models/data.

A recommend approach for us to obtain this information across the competencies of NAVAIR is to:

1. Identify the “objects” for each competency and/or aircraft domain
2. Define the associations, as shown for requirements in Figure 15
3. Define the integrated data model in the form of an ontology, so that we can leverage Semantic Web Technologies, which provide the tool agnostic approach for analysis of dependencies, assessing measures of consistency and completeness

With the new proposed framework, there needs to be some type of decision framework for assessing maturity (as Dave suggested removing the notional baselines). We plan to address this using the Information Model (as a means of collecting relevant information, and using completeness and consistency checks much like NASA/JPL).
Figure 15 provides a perspective on some of our 2016 deliverables. We have a requirement ontology, and developed and associated requirement manager prototype, which uses semantic web technology for checking requirement consistency and completeness, as shown in Figure 16. This capability was a high priority as part of the draft POA&M, but has moved lower in priority as described earlier. On the left side of Figure 16 is the requirement ontology that we’re evolving, and on the right side is a simple GUI that we have used to enter requirements, which are associated with a Telepresence Robot system that we use in a course at Stevens. It is simple, but it has many facets of interest:

- Hardware: mechanical, electrical components
- Software
- System of system (distributed)
- Sensors
- Multiple processors
- Communication, uses Wi-Fi and internet
- Humans-in-the-loop
- Mobile
- Semi-autonomous
- Needs to address availability, data integrity, safety and security
Figure 16. Ontology and Requirement Manager Engine Prototype

The next steps are to develop completeness and consistency checks for requirements, and evolve that to support risk assessment per the plan shown in Table 1.

4 Task 2 – Model Integrity – Developing and Accessing Trust in Model and Simulation Predictions

Model integrity, from our sponsor’s perspective, is a means to understand margins and uncertainty in what models and associated simulations “predict” or in other words when/how do we trust the models. The objectives characterized by the sponsor are to ensure that the research covered the key objectives; those objectives included:

- Include both models to assess “performance” and models for assessing “integrity” such as:
  - Performance: aero, propulsion, sensors, etc.
  - Integrity: Finite Element Analysis (FEA), Computational Fluid Dynamics (CFD), reliability, etc.
- A stated challenge was: how can “integrity” be accomplished when the current situation involves federations of models that are not integrated?
- Continuous hierarchical and vertical flow enabled by models and iterative refinement through tradespace analysis, concept engineering, and architecture and design analysis

Sandia National Laboratory discussed 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 and Uncertainty (QMU), an enabling framework called Design Analysis Kit for Optimization and Terascale Applications (DAKOTA) Toolkit [116], and the need and challenge of Model Validation and Simulation Qualification [113]. 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). 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.

All of these approaches remain in scope of our research, but they have been pushed out in time to address the priorities of POA&M. However, research advised under Mark Blackburn by PhD Col. Timothy West [129] at Stevens Institute of Technologies involves a proposed methodology to use the Sandia National Laboratory (SNL) DAKOTA Toolkit [116] with the Department of Defense (DoD) Computational Research and Engineering Acquisition Tools and Environments (CREATE) Air Vehicle (AV) [105] family of computational tools (e.g., CFD), in order to develop an optimized wind tunnel campaign for two different aerodynamic shapes to assess the process.

Traditional approaches referred to as Verification, Validation and Accreditation (VV&A) of modeling and simulation capabilities are still relevant and used by organizations. 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 [52]. The word tool qualification [53] and simulation qualification [113] have also been used by organizations regarding the trust in models and simulations capabilities.

See also Section 7.3 on “Model Validation and Simulation Qualification,” which include topics highly relevant to Model Integrity.

5 Task 3 – Modeling Methodologies

Task 1 and the SSTT is a study about what information/data is needed to understand both the problem and design space. Task 2 is about understanding the “thrust” in the data that is produced by various types of models and tools. Task 3 is about what are the needed and best methods to systematically produce that information. Often there is a symbiotic approach to assess the methods and select supporting tools. There have been extensive discussions about a broad spectrum of tools documented in the RT-141 final report [19]. However, there have been numerous meetings, research papers, even presentations by representatives of companies that sell modeling tools that all describe that it is critical to do several things before “buying tools.” For example, Matthew Hause who works for a company that sells MBSE modeling tools and other related products provides a list of things NOT to do when adopting MBSE. A key point from the list involves the need for organizations to: 1) understand what they need to produce (with models), and 2) the method for using the tools. Therefore, this section discusses the needs for methods as discussed in Section 2.1.

Our team is aligned with many of these “better practices,” towards MCE adoption:

- Identifying information that is needed (by NAVAIR) that is produced or analyzed through models to support decision making, see Section 3
- Methods that need to be used to enable the modeling tools to work in a more efficient manner
  - One such failing of utilizing a proper method was pointed out by the DARPA META project program manager [7]
- We need to increase focus on cross-domain methodologies to ensure tool usage produces complete and consistent information compliant with information captured in the SSTT
We also want to embed methodological guidance in the new tooling, such as: design patterns, reference models, etc.
- Methods could be represented as an SysML Activity Diagram (type of flow chart) that shows the process flow, and data flow (both in and out) to the System Model (SSTT)
- NASA/JPL provides a good example [8]

5.1 **MODELING EXAMPLES OVERVIEW**

Our sponsor requested that we develop some example reference models, MDAO models, and mission and system level models to improve the knowledge of the NAVAIR team. Per their recommendation we started with SysML characterizing capability, operational, structural, and behavioral views. The need is for people to be able to read and have discussions about the models (not necessarily be able to create the models – at least at this point). We selected some UAV scenarios and created models showing a few different types of modeling views. The following is an overview of some of the information shared; more details are provided in Section 5.1:

- Activity diagrams to describe different process models
  - Pre-modeling guidelines
  - Example CONOPS
  - Simple MBSE process, including MDAO relationship
  - Functional Requirements Decomposition
- Package hierarchy for structuring and organizing model information
  - For example, we organized this model to include:
    - Enterprise models
    - Reference models
    - Mission models
    - System models
    - Aircraft system hierarchy
- Mission level models
  - Created these model views to set the context for the system (best practice guideline)
  - Use Case diagram for mission using (Observe, Orient, Decide, Act) in context of Find, Fix, Finish
  - Activity diagram of Mission Activity relating a Sensor Platform (UAV) and its interactions with Communication Platform(s)
- System level models
  - Illustrate the system context using a Block Definition Diagram (BDD), which shows the element (systems, actors, environment) in the Mission Domain
  - Top-level Use Case for a UAV (fly, surveillance, refuel, on-ship refueling)
  - State machine diagram of the states of a UAV, from off, taxi, takeoff, cruise, loiter, descend, hold, land, etc.
- Activity diagram of Dave Cohen’s framework process
  - We expect that modeling the framework will help in supporting analysis of the challenges and gaps matrix
  - This forced use into the need to model KSA as a BDD
- Illustrated how SysML models can relate to other models using artificial MDAO parameters and constraints to model a workflow to reflect some cross-domain analysis related to Weight, Aerodynamics, Propulsion, and Performance (e.g., vehicle range) as shown in Figure 17. This example allowed us to discuss the notional steps in MDAO (For additional details see Section 8.7):
A MDAO analysis is defined as a sequence of workflows (scenarios)
- After identifying a set of inputs and outputs (parameters)
- Define a Design of Experiments (DoE) and use analyses such as sensitivity analysis and visualizations to understand the key parameter (this scopes the problem)
- Use Optimization using solvers with the key parameters and define different (key objective functions – on outputs) to determine set of solutions (results often provided as a table of possible solutions)
- Use visualizations to understand relationships of different solutions
- NOTE: Any node on an Architectural (SysML) model could map to some Physics-based model
- Some of these architectural views from SysML models can be workflows of analysis through MDAO (using Magic Draw, and Rhapsody)

- We followed-up this session by showing some MDAO webinars during a working lunch, we showed and discussed
  - “Part III: MDAO for Conceptual Aircraft Design at Northrup Grumman”
  - Provided links to other relevant Webinars from different contracting organization to NAVAIR
    - System Trade Studies & Design Optimization, presented by Lockheed Martin
    - Phoenix Integration (ModelCenter) & the Skunk Works presented by Boeing
    - The Role of Multi-Domain Dynamic Models for Functional Verification in MBSE
  - Link is here to more webinars: http://www.phoenix-int.com/resources/webinars/ondemand-webinars.php

### 5.2 Modeling Examples

The SSTT is an enabler for cross-domain interoperability needed for MDAO problem and design space exploration. The SSTT requires that all information used to assess performance is semantically consistent with MCE technologies and methods used for assuring integrity and the orchestrated workflow is data-driven (not process driven). This concept will be applied with an ontology-driven reasoning framework to
provide a more objective assessment of the maturity of a design as it provides value to the three KSAs, risk, and uncertainty.

This section provides some examples of the models shown during the 17-August working session.

### 5.2.1 Table of Contents

We created a Content Diagram as a type of table of contents where we put hyperlinks to the different diagrams. This section shows some of the diagrams in each of these groups.

![Content Diagram](image)

**Figure 18. Table of Contents to Models and Diagrams**

### 5.2.2 Process/Methods

We created a few SysML activity diagrams to describe different process/methods models. Ideally, these types of models would be reference models that establish a best practice approach for starting different types of models. These guidelines are shown in an activity diagram called Pre-modeling guidelines. Such guidelines as shown in Figure 19, include defining the structure of the overall project, naming conventions, colors. This picture also shows the different types of SysML diagrams. In the remainder of this section we will show a few diagram types. The pre-modeling guidelines are defined in an activity diagram, which shows the actions, control flows, and can show data flows. The dark bar can represent a fork (to distribute asynchronous processes) or a join to synchronize distributed processes. One of the first things a team needs to decide is on the structure of the overarching model. An example is shown in the
Containment view as shown in Figure 20. We used the MagicDraw tool, but SysML is a standard modeling language supported by many different tools.

Figure 19. Pre-modeling Guidelines
Figure 20. Containment Structure

Figure 21 shows an activity diagram that was developed under the Stevens Institute of Technology SYS-750 Advanced Architecture Course. This activity diagram illustrates an overarching, but simplified MBSE process. This activity diagram shows the control flow (without feedback), but also the data flow to the objects colored in blue. These objects highlight the types of diagrams that might be used to capture the information in the various steps of the process. These activities can be further decomposed to describe additional details of the process steps and other information that is used or produced.

Note also that there is a Perform MDAO activity, colored orange Figure 21. This type of analysis is performed by other types of modeling tools. This is a linkage to modeling and simulation of cross-domain disciplines such as the workflow illustrated in Figure 17 that included, Weight, Aero, Propulsion, Performance, etc. There are some tools that allow the parameters and constraints to be specified in SysML and then integrated into MDAO tools such as ModelCenter, which is shown in Figure 17.

These types of activity diagrams that represent processes or methods are not representing the target system, but can be characterized as reference models used as part of the Enterprise that includes what we have referred to in our prior technical reports as the “Designing System” [19].
5.2.3 Package Hierarchy for Structuring and Organizing Model Information

We showed an example of a package structure for organizing the different perspective (enterprise, mission, system), but we can also use a similar concept to organize the actual system structure, from either a logical or physical perspective, see Figure 22. Model Organization Figure 22. This is a good decision to make by the team as early as possible.
5.2.4 **M I S S I O N L E V E L M O D E L S**

Mission-level models set the context for the “system of interest,” which is a typical best practice. From a military perspective, we included a Use Case diagram for mission using (Observe, Orient, Decide, Act) in context of Find, Fix, Finish. This could be a reusable Use Case, where the specific textual elements of the use case could be tailored to a specific mission as shown in Figure 24. We also provided an example of Activity diagram of Mission Activity relating a Sensor Platform (UAV) and its interactions with Communication Platform(s) as shown in Figure 25 [124]. Note that this concept is presented from a logical perspective and shows both control flow (dash lines), and data flow (solid lines); this activity diagram also shows swim lanes that illustrate the different partitioning of the activities. NOTE: these are all examples.
Figure 23. High-Level Mission Use Case
Figure 24. Textual Element of the Use Case
5.2.5 SYSTEM LEVEL MODELS

Use Cases are also a common starting point for system level operational scenarios. An example top-level Use Case for a UAV (fly, surveillance, refuel, on-ship refueling) is shown in Figure 26. Again, each of the use cases would typically have a structured narrative created using a template similar to Figure 24. Therefore, when we discuss using models, we do not imply that there is no textual narrative, rather the narrative is often structured and embedded within some type of modeling element.
To illustrate another type of behavioral perspective, we showed an incomplete state machine diagram of the states of a UAV, from off, parked, taxi, takeoff, cruise, loiter, descend, hold, land, etc. as shown in Figure 27.

Finally, to illustrate the generality of the SysML modeling approach, we created a model of Dave Cohen’s framework (shown in Figure 4). We expect that modeling the framework, as shown in Figure 28, will help in supporting analysis of the challenges and gaps matrix (see Figure 5). This forced us into the need to model KSA as a SysML Block Definition Diagram (BDD).
5.3 VIEWS AND VIEWPOINTS

The working session had a section on explaining various different methods for doing system engineering modeling. This also relates to the concept of View and Viewpoints in the SSTT [48]. Each Viewpoint, as shown in Figure 29, is a specification of conventions and rules for constructing and using a View for purposes of addressing a set of stakeholder concerns, which is based on a standard that relates to:

- Purpose of the viewpoint
- Stakeholder that are likely to use the viewpoint
- Concern of the stakeholder
- Method to develop a Model using a Modeling Language
- Analysis that can be performed with the models
5.4 **MULTIDISCIPLINARY DESIGN, ANALYSIS AND OPTIMIZATION**

Our sponsor asked for an explanation of MDAO methods and tools. As shown in Figure 17, we created an example scenario using an MDAO workflow. This section provides some additional rationale for the importance of an MDAO capability for both problem and design tradespace analysis.

### 5.4.1 HISTORICAL CONTEXT FROM INDUSTRY DISCUSSIONS ON MDAO

Many organizations from our RT-48/118/141 discussions use MDAO. Organization now have established the information technology infrastructure to facilitate the integrations across the design space 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 integrated across-domains in real-time. They are systematic about creation of design of experiments. Organizations stated that these technologies often uncover or expose things that are not intuitive in the design. The power from the automation and efficiency of the tools often allows them to spend more time doing more in-depth analysis; they stated that they often do 100 or 1000 times as many excursion of the design space with MDAO version traditional manual methods.

Figure 30 provides a comparison of a Legacy approach to Multi-disciplinary Design Optimization (MDO) analogous to MDAO:

- Setup time is longer (specification)
- Execution time is shorter
- Management time is significantly shorter
- Reasoning time is longer

However, what is critical is that once the experimentation framework was setup, this particular group was able to do 500 times as many iterations to examine various different alternatives in the same amount of time. We heard similar stories during our organizational discussions that were using MDAO. This supports data-to-decisions with engineering technical data and information that is produced, not just documents. The point continually made by Jaime is that the number of added excursion provides greater quality and certainty about the potential design, thus reducing risk.
5.4.2 MDAO-relevant METHODS AND TOOLS

This section summarizes the material that we discussed in a working session, which includes:

- Problem formulation method, which usually involves
  - Identification of design variables, constraints, objectives, and models of the disciplines
  - Solvers, constraints/objectives, outputs to analyze relationships between inputs and outputs, given constraints and objectives
    - Consider weighting based on importance of inputs/outputs/objectives
    - Local variables (used by one discipline), shared (used by multiple), which can create dependency relationships such as the variable \( y_2 \) in the equations in Figure 31
  - Optimization means to find a set of system configurations (input variables) that meets the objectives and satisfy the constraints
  - Problems can be multidiscipline and multi-objective
- Tools – open source and commercial (non-exhaustive) \(^{6}\)
  - DAKOTA, OpenMDAO, iSight, ModelCenter, modeFRONTIER, FIDO, IMAGE, CONSOL-OPTCAD
  - Sometimes referred to as Process Integration and Design Optimization (PIDO)
- Modeling and simulation inventory analysis
  - We were provided with an inventory analysis of the modeling and simulation capabilities at NAVAIR – there were 348 identified, most of these were for mission-level simulation, but it was believed that there are many more used by the competencies
- Need to elaborate on the MDAO methods
  - An MDAO analysis is defined as a sequence of workflow (scenarios)
  - After identifying a set of inputs and outputs (parameters)

---

\(^{6}\) Certain commercial software products are identified in this material. These products were used only for demonstration purposes. This use does not imply approval or endorsement by Stevens, SERC, or NAVIAR, nor does it imply these products are necessarily the best available for the purpose. Other product names, company names, images, or names of platforms referenced herein may be trademarks or registered trademarks of their respective companies, and they are used for identification purposes only.
- Define a Design of Experiments (DoE) and use analyses such as sensitivity analysis to understand the key parameter (this scopes the problem)
- Use Optimization using solvers with the key parameters and define different (key objective functions – on outputs) to determine set of solutions (results often provided as a table of possible solutions)
- Use visualizations to understand relationships of different solutions

Part of new contracting, should potentially require contractor to provide methodology so that NAVAIR can interpret and understand analysis data and results

Figure 31 shows what would be considered an example of a canonical problem and MDO “Architecture” based on Design Structure Matrix. We discussed another example where we stepped through how the variables, input, outputs, objective (e.g., min F), variable dependencies and constraints are represented in this general structure. Solvers are selected based on differ variable that are related to different domains.

\[
\begin{align*}
given & \quad y_1 = D_1(x_1,y_2,z_1,z_2) \\
y_2 &= D_2(y_1,z_1,z_2) \\
\min & \quad F(x_1,y_1,y_2,z_2) \\
w.r.t. & \quad x_1,y_1,y_2,z_1,z_2 \\
s.t. & \quad H(y_2,y_2) = 0 \\
& \quad G_1(y_1) \geq 0 \\
& \quad G_2(y_2) \geq 0
\end{align*}
\]

**Figure 31. Canonical Problem & MDO “Architecture” based on Design Structure Matrix**

We then discussed how Figure 31 might be more generally represented in a tool. Jaime was interested in OpenMDAO, which was started as a development by NASA. The typical pattern of elements for OpenMDAO are shown in Figure 32, which partitions the problem in different workflows that might represent analyses across different domains. OpenMDAO uses python code as a means of constructing this type of setup, but currently does not have a graphical user interface (GUI). A notional scenario is that “driver 2” is some type of structural analysis (finite element analysis [FEA]) and “driver 3” is some type of aerodynamics (computational fluid dynamics [CFD]) and “driver 4” is some type of sensor simulation related to the aerodynamics, and the main “driver” integrates the multiple discipline simulations to look at the cross-cutting aspects of aero, structures, and sensors.
We next discussed a number of other commercial tools, which often do have a GUI and often contains many other capabilities, such as visualization, graphic workflow construction, design of experiments, and integration with other tools and solvers.

5.5 **CAPABILITY AND OPERATIONAL-LEVEL MODELING GUIDELINES**

NAVAIR has an architecture group that constructs models associated with the Capability Description Document. They have defined guideline to provide some methodological guidance in constructing DoDAF models. Notionally, the guidelines cover: capabilities view, operational views and system views. These are generalized as architecture models.

- The “architectural modelers” who create DoDAF based on these guidelines use the UML Integrated Architecture (UPIA) profile; they have some notional “ontologies”
  - Integrated Operational Model Ontology
    - Capability Ontology (WP 1010-0000-01)
    - Operational Ontology (WP 1020-0000-01)
  - Integrated System Interface Model Ontology
  - System Interface Ontology (WP 1030-0000-02)
  - Diagram Ontology (WP 1060-0000-01)
  - Operational Diagram Ontology
    - OV-5a Use Case
    - OV-5b Activity Diagram
    - OV-6c Event Trace
  - System Diagram Ontology
    - SV-4a Use Case
    - SV-4b Activity Diagram
    - SV-10c Event Trace
Currently, these efforts are fundamentally limited to the net-ready aspects of capabilities, governing communications and interoperability. It is acknowledged that extending this to logical and functional views is desirable for the SE transformation.

5.6 NAVAIR STUDY VIEWS

Study views were created to address a number of challenges at the Program of Record (POR) level and in creating DoDAF requirements, however there does not appear to be extensive knowledge about their use. 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, which is 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 dynamics needed for analysis.

A mission-level SoS analysis begins with formalization using Study Views, as reflected in Figure 33, which has modeling and simulation 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 by the NAVAIR Architecture group (see Section 5.5). This information forms the analysis boundaries for the System Capabilities information needed to define requirements for the POR.
5.7 MODELING AND METHODS FOR UNCERTAINTY QUANTIFICATION

As discussed in Section 4, Sandia National Laboratory discussed some advanced methods 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 and Uncertainty (QMU), an enabling framework called Dakota, and the need and challenge of Model Validation and Simulation Qualification [113]. 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). 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.

---

7 Image source: Thomas Thompson, Enabling Architecture Interoperability Initiative, B210-001D-0051 Unclassified.
5.7.1 Dakota Sensitivity Analysis and Uncertainty Quantification (UQ)

The Dakota framework supports optimization and uncertainty analysis [116]. 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 come 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 representative of the concept of model-centric engineering, see Figure 34. It automates typical “parameter variation” studies to support various advanced methods 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)
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.

This approach and Dakota framework 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

### 5.7.2 AN OVERVIEW OF QUANTIFICATION OF MARGINS AND UNCERTAINTY

Dakota is a tool framework that can support the method of Quantification of Margins and Uncertainty (QMU). Some of the material from Sandia is categorized “Official Use Only [OUO].” We provide a summary extracted from publically available information [94].

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\(^8\). 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 [94]. We believe the methodology has more general use than just physics package certification.

QMU applies to the lifecycle of the entire 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

\(^8\) 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
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 [94], which:

- 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 [94]
  - 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).
The information is relevant to the risk framework as it provides evidence about methodologies and tools to deal with several of the topics. 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

5.8 MODELING METHODS FOR RISK

The risk modeling and analysis methods address potential errors and uncertainties in the overuse of limited 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 help deal with these type of issues. This particular example uses a Bayesian model [110].

5.8.1 PREDICTIVE MODELS FOR RISK

There are situations where we do not have good historical quantitative data and we often use expert judgment. This section discusses 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 [50] 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 [15][16][17]. 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](image)

5.9 **Controlled Natural Language Requirements Information**

We acknowledge that there may not be value in modeling everything, and believe a risk-driven approach to modeling should be considered. In addition, if the concepts associated with the vision are valid, subject matter experts using a rich web view (see in Figure 9) may want to supplement models with other types of constraints. We discussed in working session research into Controlled Natural Language (CNL)
Requirements and ontology-driven Natural Language Processing of requirements. The fundamental premise is that we can structure textual-based requirements that we can then use automated means to formalize the requirements for analysis of consistency and completeness in the context of an ontology; there have been a number of different research efforts that have demonstrated the successful transformations of CNL requirements into an analyzable form. In addition, we also believe that while we will transition to the use of models, there will always be subject matter experts that will augment the representation from models with constraints using some form of textual-based specification. Note also, that in our industry visits there were two organizations that explicitly discussed requirement generation from models, discussed as part of the DARPA META project, and has been discussed by the Engineered Resilient System research.

- **Purpose for CNL**
  - Constrains the way requirement statements are constructed
  - Supports tool-based analysis
  - Improves consistency
  - Allows for template-based generation of formalized and analyzed requirements
  - Can integrate with Rich Modeling in SSTT

- **Approach example – used by both Lockheed Martin C5 (and presented in open forum)**
  - Goal: specify the behavior of the outputs in terms of the inputs
  - Use limited set of action verbs combined with structured, repeatable phrasing (syntax) for requirements, and improve understanding between the specifier and developer/reviewer
    - Eliminates confusion caused by multiple terms used for the same purpose
    - Examples: derive vs. compute vs. calculate vs. determine vs. process ...
    - All of these essentially mean “execute the logical/algorithmic steps to set the output based on the input(s).”

- This provides a pattern for analysis and development of requirements
  - **Provide** requirements define the outputs
  - **Derive** requirements specify the algorithms for producing the terms which are output
  - **Acquire & Validate** requirements identify the input signals needed to derive the terms
    - Definition of action verbs helps ensure all issues get addressed
    - Validation of Input
    - Error Handling
    - Source/Destination Specification

- **Example**
  - DATA ACQUISITION:
    - Mission Processing System shall acquire <alias>.
  - DATA VALIDATION:
    - Mission Processing System shall perform data validation on <alias> per Table <table-id>.
    - Mission Processing System shall set <validity_alias> to <enumeration> when <all|any> respective data validation checks in Table <table-id> <pass|fail>.

- Has been developed using a spreadsheet to control structure, verbs, etc.

## 6 Task 4 – Define System Engineering Transformation Roadmap

Our plans for the roadmap aligned with the prioritized set of goal shown in Table 1, which also highlights traceability to Technology, Methods, and Competencies. The expected development of a roadmap focused on transitioning from the traditional SETR approach, to a requirement, risk, and evidence-based
approach using an evolving underlying SSTT. We planned fourth to focus on MDAO to be more systematic in tradespace of the problem space. This was also focused at the POR level.

However, the acceleration of the SE transformation as altered that plan. We are now working our gaps and challenges associated with the new framework (see Figure 5), which include, but are not limited to:

- Using an interactive approach to MDAO in a collaborative effort to develop a new type of specification, RFP & SOW (Steps 1-4 in Figure 4)
- Developing strategies to track and assess value of requirements to KSA
- Investigating a collaborative operational paradigm between government and industry
- Accelerating the awareness of modeling methods for the competencies (see Section 5)
- Considering alternative digital engineering strategies for evaluating a proposal

In support of the new operational paradigm, we were requested by our sponsor to attend a workshop to discuss strategies for how digital models could be used in the DoD acquisition process to support competitive down-selection for:

- Competitive prototyping in the Technology Maturation and Risk Reduction phase, and
- Engineering and Manufacturing Development.

The objectives of the workshop were:

- Obtain community input on the types of digital models (design, cost, performance, mission, etc.) that could be used to support competitive down-selection.
- Identify the existing gaps that must be closed to enable the use of digital models to support competitive down-selection.
- Recommend changes in the DoD acquisition process and Government-Industry interactions to enable the use of digital models to support competitive down-selection.
- Obtain community input on the policy and legal issues associated with the use of digital models in Requests for Proposals (RFPs), proposals, and during proposal evaluation to support competitive down-selection.

Therefore, as part of the newly awarded RT-170, we will update the roadmap to include the new task as discussed in Section 2.5, and consider topics discussed in working sessions:

- Virtualization and event-driven reviews
- Implications of a Computationally Enabled System Engineering through the formalization of the Decision Framework
  - Concept to embed System Engineering through cross-domain linkages and relationships in the Information Model that underlies ISEE
  - Can we measure requirement stabilization?
- Significant discussion about the implications of Software Intensive Cyber Physical System (CPS)
  - Insights gained from working with JSF
  - Revisit RT-142 on Risk Leading Indicator on SW-intensive CPS
- Models enable new possibilities for verification at proposal evaluation
- Collaborative approach to decisions-in-the-loop (Step 5-7 of framework)
  - Measure of design maturing
- Integrated System Engineering Environment (ISEE)
  - Variant management, which relates capturing tradespace analysis
  - Template-based approach to requirement generation (and possibly contract generation)
- Formalizing contracting
  - Handoff between Steps 4 and 5 of framework
Examples include contracts specification language (GCSL) developed by the Designing for Adaptability and evolution in System of systems Engineering (DANSE) project

- Make Contract Data Requirements List (CDRLs) about Verification and Validation (V&V) and Model Integrity
- Early planning for sustainment
  - Digital Twin – “An integrated, multiphysics, multiscale, probabilistic simulation of an as-built system, enabled by Digital Thread, that uses the best available models, sensor information, and input data to mirror and predict activities/performance over the life of its corresponding physical twin.”
- Risks
  - Airworthiness and Safety (most critical in Technical Feasibility assessment)
  - Program execution (cost, schedule and performance)
  - Competencies may not be ready for the first pilot (see Figure 1)

7 **INTEGRATED FRAMEWORK FOR RISK IDENTIFICATION AND MANAGEMENT**

The research under RT-48/118/141 described strategies, methods, and tools for a risk-based framework that aligns with the Vision model concept through MCE. This involves how the Vision model should include integrated risk identification and management. Many of the topics in this section could have been included in the either the model integrity (see Section 4) or modeling methodologies section (see Section 5). We believe that many of these topics are still potential challenges in the new operational paradigm characterized by the SE transformation framework. Many of the topics are fundamental the systems engineering, and are grouped into this section in this interim report.

7.1 **RISK CONTEXT**

Defined in the DoD Risk Management Guide [51],

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, and
- 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.
7.1.1 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 [87]), 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 [46]. 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 and 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 (non-linearities).

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 from 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 [85], validation and accreditation for MCE with respect to airworthiness and safety may require significant effort and expertise [52]. 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.

7.1.2 FUTURE ROOT CAUSES

As the focus of the effort is on understanding the problem, 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.
- 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.

7.2 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 [80]. 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)

### 7.2.1 Risk Framework Captures Knowledge

A risk framework is actually a knowledge model of credibility (not models of performance, but models of uncertainty). 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:

- 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 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. However, if internal model parameters can be adjusted in this manner, this means that there is some amount of uncertainty associated with the true, or best, values of these parameters. Uncertainty associated with model inputs directly implies uncertainty associated with model outputs [84].

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 [110], 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 [113]
- “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 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 [113]. 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. In addition, Sandia discussed the Predictive Capability Maturity Model (PCMM), which is an evolving model that can be used to assess the level of maturity of computational modeling and simulation (M&S) efforts [95]. The PCMM addresses six contributing elements to M&S: (1) representation and geometric fidelity, (2) physics and material model
fidelity, (3) code verification, (4) solution verification, (5) model validation, and (6) uncertainty quantification and sensitivity analysis.

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 [62]**
  - 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**
  - This was 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**

- **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 [83]**
  - 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

- **Bayesian-based qualification planning [110]**

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

## 7.4 Improving the Integrity of Models

The DARPA META program discussed the issues and therefore risks with attempting to use models that were not developed according to a methodology that results in models that are suitable to the capabilities of the tools [7]. Increasingly, there are approaches being used to improve the integrity of the models through various types of checks, for examples:
- NASA/JPL use an ontology-based approach for ensuring models are developed to comply with modeling patterns, and have developed over 60,000 checks using computationally based formal reason to ensure the models are well-formed, consistent and complete with respect to the modeling pattern [74]
- Formalized evaluation criteria of a modeled system architecture by encoding 23 axioms in two different tools: Vitech CORE and Innoslate from SPEC Innovations to demonstrate concept [111]
  - Some tools provide pre-defined checks, but in this case the tools allow for additional types of checks to be performed that can be adapted to a particular methodology
- Apply an approach for Verification, Validation, and Accreditation Shortfalls for Modeling and Simulation based on criteria associated with modeling and simulation patterns in SysML [103]
- DARPA META also developed a concept of Probabilistic Certificate of Correctness (PCC) [7]
  - PCC is used to verify that the system will perform within the requirements bounds within a probabilistic certainty factor
    - PCC uses statistical sampling techniques on system and environmental parameters, executed on test benches for evaluation of the statistical properties of the metrics
    - Requirements objectives and thresholds define the numerical bounds of the acceptable metric values, and a probability of the design metric existing within the bounds is computed
  - Variation in performance is expected due to manufacturing variations of components and possibly environmental conditions

A risk framework needs to consider these and other types of checks. These types of checks not only ensure that the model is well-formed, consistent and complete with respect to modeling patterns, but this also reduces risk, because these types of rules provide greater assurance from the use of tools that use these models for simulation, analysis and generation.

### 7.5 Model-Centric Methods and Tools Enable Approaches to Safety and Airworthiness

The emerging approach to Failure Modes and Effects Analysis (FMEA) and safety analysis is to use formal, automated methods to derive the FMEA and safety analysis models from the SE model of the system, instead of the historical practice of developing informal models by manual means, and to be able to integrate failure and safety analysis earlier in design. The reliance on the SE model of the system potentially exposed the failure mode and safety analysis to the risk of an incomplete or incorrect systems engineering model. This risk can be mitigated by:

1. Using formal methods based on a meta-model in developing the system model
2. Applying automatic model verification methods to the system model
3. Applying error and uncertainty propagation methods to analyze the accuracy of the system model

These are areas of active research and development, and the methods, tools and techniques are rapidly maturing. The current state-of-the-art is that there are available methods with demonstrated capability, but not complete capability or entirely turnkey operation. Research and development is focusing on integrating software FMEA with hardware FMEA, and higher-level resolution of failure modes.

We believe that formalization of MCE can lead to advances aligned with the following hypotheses:

- H1. Model-based methods to automatically generate Failure Modes and Effects Analysis (FMEA) models tools (1) are feasible and practical, (2) are superior to traditional manual methods, (3) can address software failure modes, and software-hardware interactions in cyber-physical systems.
- H2: Model-based safety analysis methods, procedures and tools (1) are feasible and practical, and (2) are suitable for analysis of mission critical requirements and design in the phases of the acquisition lifecycle prior to flight certification testing.
- H3: Risks in reliance on models are manageable, and model analysis methods provide a level of assurance of model correctness and accuracy.

We provide a summary of some research finding in support of the hypotheses.

7.5.1 MCE METHODS TO GENERATE FAILURE MODES AND EFFECTS ANALYSIS (FMEA) MODELS

Model-based methods to automatically or semi-automatically generate Failure Modes and Effects Analysis (FMEA) models from Systems Engineering (SE) models of requirements, function, design and their interaction are feasible, practical, and have been demonstrated. There are alternative technical approaches to enhancing the system model to automatically identify abnormal modes and chains of effects. The automated approaches do not address the criticality of different system functions, only how the functions are degraded. The automated modeling approach provides higher resolution of degraded mode states for high-level functions as a combinatorial analysis of low-level failure states. The SE models need to conform to a meta-model framework for automating FMEA models.

Schindel describes a systematic approach beginning with a “metamodel” to organize and express the system requirements and high-level design MBSE model [117]. The MBSE model defines the system architecture, functional requirements and functional dependencies. The system architecture is the organization of the system into subsystems down to a “black box” level, and the interconnections or links between elements. Functional requirements interaction behaviors and performance under specified conditions in an objective technical language for the system and every subsystem down to the “black-box” resolution of the MBSE model, and for each of the links between elements. Conditions are external and internal states of the system and its internal subsystems. Behaviors are state changes. Functional dependencies are the hierarchical relationships among functions, in which higher-level functions depend on lower level functions. Requirements are automatically converted into counter-statements that define the failure modes – failures of system elements and links to perform one or more of their functions. The effects are automatically traced through the links and nodes in the system architecture, and propagated up through the functional hierarchy.

Increasing complexity makes manual methods laborious, and highly depends on the experience and expertise of the practitioners. Manual approaches are subject to human error of omission, and are not tightly linked to the system model. These issues are addressed with automated FMEA model generation from the Systems Engineering model. Automated FMEA model generation creates the possibility to extend FMEA to a comprehensive analysis of degraded mode states. The automated methods do not distinguish between important (likely and significant) failure modes in developing the FMEA model – a process guided by the insight and experience of the practitioners in manual FMEA modeling. The FMEA models generated by automatic methods are too large to understand by reviewing the model. Automated analysis methods are needed. These methods are still under development. Automated analysis requires input data regarding the likelihood or frequency of the different failure modes, and the importance of high-level function degradation mode. Manual models require the same data, but since the data are often not available (just subjective estimates), the dependencies are built into the model based on “expertise and experience.”

Automated FMEA analysis can address software failure modes, and software-hardware interactions in cyber-physical systems. FMEA addresses the behavior and response of the system to changes in (a) external conditions (environment, demand), and internal conditions (performance time, accuracy and
error of functional processes and the state or mode of logical subsystems). State transition modeling is a “best practice” to specify the system behavior and to test the behavior specifications. Automated FMEA for software requires a model of the logical states and transitions of the information in the system, as well as the physical states and transitions of the system and its environment. In real-time cyber-physical systems, timing of effects of sensing, processing, signaling and response on transitions needs to be included.

There are many possible failure causes and modes. The nominal behavior as specified in the state transition model may not be the desired behavior, i.e. the conceptual design may not satisfy the requirements, or can exhibit unplanned behaviors (“sneak circuits” and unplanned feedback loops). The behavior specifications (the state transition model) may be incomplete, i.e., there are possible states and transitions that are not represented in the model, e.g., the effects of failure of an actuator (stuck or random response), noise or failure of a sensor, short-circuits, and unexpected combinations of logical information states. The system can have calibration errors, or not account for effects of temperature, pressure, etc. on the calibration. Discretization of time and signal level can lead to control systems exhibiting behavior different than expected from a continuous model. Software specifications can be implemented incorrectly, i.e., coding errors.

Snooke presents an approach for automated, model-based software FMEA [121]. He addresses three classes of software failures: abnormal input values to a routine, failure of the processing hardware, and logical/algorithmic/semantic error in the software code. He does not address incomplete or incorrect specifications for the software. His approach combines methods from software diagnosis (functional dependency modeling) with automatic fault tree methods.

Another approach for automated, model-based software FMEA demonstrates the method in an application to aircraft flight control software [134]. The approach is concerned with failure modes resulting from the state of logical information in the system and the transformation of information state by processing, including timing constraints and synchronization. The approach relies on a system requirements model of behaviors and performance under conditions, and a software requirements model of information states, state transformations, timing interactions, etc. They define a structured approach to system and software modeling, and for the use of the models to analyze software failure modes and effects. They applied the method to flight control software that controls the cabin door, the brake system and the front flap.

7.5.2 MISSION-LEVEL MODEL-BASED SAFETY ANALYSIS

Model-based safety analysis methods, procedures and tools are feasible and practical for mission critical safety assessment. System safety analysis methods are commonly based on informal system models. The lack of precise models of the system architecture and its failure modes often forces the safety analysts to devote much of their effort to finding undocumented details of the system behavior and embedding this information in the safety artifacts such as the fault trees.

NASA/JPL developed a comprehensive Safety-Driven, Model-Based System Engineering Methodology that enables system engineers to design systems from a safety point-of-view, i.e., with hazard analysis folded into the nominal design process rather than conducted as a separate activity [65]. This methodology integrates MIT’s Systems-Theoretic Accident Model and Processes (STAMP), STAMP-Based Hazard Analysis (STPA), intent specifications (a structured, constraint-based system engineering specification framework), and JPL’s State Analysis (a model-based systems engineering approach). The methodology was developed to address the challenges of safety design and analysis for complex, software-intensive cyber-physical systems, and to do so early in the design process.
Model-based safety analysis has been approached as a special case of automated, model-based FMEA approaches, except that instead of addressing all system requirements and their derived requirements, it is restricted to safety-related requirements and their derived requirements. Benefits of model-based safety analysis include:

- Tight integration between systems and safety analysis based on shared models of system architecture and failure modes
- The ability to simulate the behavior of system architectures early in the development process to explore potential hazards
- The ability to exhaustively explore all possible behaviors of a system architecture with respect to some safety property of interest using automated analysis tools
- The ability to automatically generate many of the artifacts that are manually created during a traditional safety analysis such as fault trees and FMEA/FMECA charts

Joshi and Heimdahl leverage existing tools and techniques from model-based development to create formal safety models using tools that are familiar to engineers and the static analysis infrastructure available for these tools [75]. They enhance the system model with a fault and effects model to automate much of safety analysis, and do so in a way that the safety issues point back to design elements that produced the adverse behavior. Their approach composes a model of nominal system behavior to model of failure modes and effects on behavior under a set of common malfunctions and failure modes of elements of cyber-physical systems. Their approach uses automated model checking methods that will either verify safe behavior in the presence of “N” faults, or find a combination of “N” faults that produce adverse behavior (i.e., failure to meet a safety critical functional requirement). They apply the method to the aircraft wheel braking system example from ARP4761 [115].

HiP-HOPS (Hierarchically Performed Hazard Origin and Propagation Studies) is an automated model-based method for safety analysis that enables integrated assessment of a complex system from the functional level through to the low level of component failure modes [101]. The failure behavior of components in the model is analyzed using a modification of classical FMEA called Interface Focused-FMEA (IF-FMEA). One of the strong points of this approach is that the fault tree synthesis algorithm neatly captures the hierarchical structure of the system in the fault tree.

### 7.5.3 Risk Reliance on Model Correctness and Accuracy

Risks in reliance on models are manageable and model analysis methods provide a level of assurance of model correctness and accuracy. In the transformation vision, models become the “language” for system development: requirements are expressed as functional models in SysML, UML or similar conventions, specifications are expressed as state-transition models (or other method of computation, as appropriate to the system), the specification models automatically produce code and component performance specifications, automatic methods generate FMEA, and high-fidelity system operation models are used for virtual testing and design of experiments for physical testing.

Using incomplete or incorrect models at any stage exposes the development program to the risk of erroneous design decisions. In the “As Is” document-centric development process, the program has greater exposure to this risk because it lacks the ability to test-and-verify the implications of requirements, specifications, and design, and because the opportunity for human error is introduced at each state. Model-centric development reduces this risk as a metamodel, as shown in Figure 38 can formalize the required information and relationships need to ensure a complete analysis. As a cadre of engineers emerges with expertise in MCE with experience using improved modeling tools and
frameworks, and model checking methods and tools, the risks from incomplete or incorrect models will decline.

Figure 38. Example Failure Modes and Effects Analysis Metamodel

Using formal modeling methods based on a meta-model will help ensure that the system models can be checked for completeness and correctness as they are being developed. Automated model checking methods procedures and tools is a growing field with commercial products and expanding scope. These tools automatically check model completeness, and, to some extent, correctness.

Model-centric development introduces a different risk – the risk of uncritical review of the modeling and analysis methods and results. In document-centric development there is usually health skepticism, and reliance on experienced subject matter experts to review the development documents. In model-centric development, DoD will need to develop a cadre of experts with expertise both in the domain and in modeling and analysis methods.

8 Part II Summary

Our research suggests that model-centric engineering is in use and adoption seems to be accelerating. As described herein, our sponsor recognized the need to make a radical transformation and are developing a strategic plan based on a new operational paradigm for acquisition and design to accelerate a SE transformation. We are adapting our research strategy and focus to align with their evolving plan. This message has been shared more broadly with SERC sponsors, government sponsors of SERC research, and industry both through SERC and NDIA events.
In a recent Government-Industry Model Centric Engineering forum conducted by the Systems Engineering Research Center (SERC) and the Office of the Undersecretary of Defense, the following four perceived areas of benefits were found to be the key themes to implementing [39]:

1. Improved Acquisition – accepting digital deliverables could improve the governments understanding of a projects status and risk along with allowing them to “validate” the contractor’s deliverables.
2. Improved Efficiency and Effectiveness – reduce time and effort in the performance of existing tasks using a digital “twin” of the system.
3. Improved Communication; Better Trade-Space Exploration; Reduced Risk – using ontology-based information models to translate and extract useful information between a variety of models and model types could allow for improved communication among specialists.
4. Improved Designs and resulting Systems and Solutions – being able to understand the impact of requirement and/or design decisions early could help improve the overall system design and identify adverse consequences of the design before committing to a design choice.

The future research of MCE will need to take into account these four perceived areas of benefit and help make progress toward these dimensions. The path forward to transitioning to MCE has both challenges and many opportunities, both technical and sociotechnical. The modeling infrastructure for a digital engineering environment is a critical step to enable a SSTT, which we believe can better link information across domains for better and earlier decision making. While there are thousands of tools they are often federated and there is currently not one single solution that can be purchased to span the MCE lifecycle. Every organization providing inputs to this research has had to architect and engineer their own model-centric engineering environment. Most have selected commercial tools and then developed the integrating fabric between the different tools, model, and data. This often uniquely positions them with some advantages among others in theirs industry. Some organizations have encoded historical knowledge in reference models, model patterns to embed methodological guidance to support continuous orchestration of analysis through new modeling metrics, automated workflow, and more. Our immediate challenge is to work the challenge areas (Figure 5) associated with the new framework Figure 4.
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.

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>AADL</td>
<td>Architecture Analysis &amp; Design Language</td>
</tr>
<tr>
<td>ACAT</td>
<td>Acquisition Category</td>
</tr>
<tr>
<td>AFT</td>
<td>Architecture Framework Tool of NASA/JPL</td>
</tr>
<tr>
<td>AGI</td>
<td>Analytical Graphics, Inc.</td>
</tr>
<tr>
<td>AGM</td>
<td>Acquisition Guidance Model</td>
</tr>
<tr>
<td>ANSI</td>
<td>American National Standards Institute</td>
</tr>
<tr>
<td>AP233</td>
<td>Application Protocol 233</td>
</tr>
<tr>
<td>ATL</td>
<td>ATLAS Transformation Language</td>
</tr>
<tr>
<td>ASR</td>
<td>Alternative System Review</td>
</tr>
<tr>
<td>AVSI</td>
<td>Aerospace Vehicle Systems Institute</td>
</tr>
<tr>
<td>BDD</td>
<td>SysML Block Definition Diagram</td>
</tr>
<tr>
<td>BN</td>
<td>Bayesian Network</td>
</tr>
<tr>
<td>BNF</td>
<td>Backus Naur Form</td>
</tr>
<tr>
<td>BOM</td>
<td>Bill of Material</td>
</tr>
<tr>
<td>BPML</td>
<td>Business Process Modeling Language</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>CASE</td>
<td>Computer-Aided Software Engineering</td>
</tr>
<tr>
<td>CDR</td>
<td>Critical Design Review</td>
</tr>
<tr>
<td>CEO</td>
<td>Chief Executive Officer</td>
</tr>
<tr>
<td>CESUN</td>
<td>International Engineering Systems Symposium</td>
</tr>
<tr>
<td>CMM</td>
<td>Capability Maturity Model</td>
</tr>
<tr>
<td>CMMI</td>
<td>Capability Maturity Model Integration</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Requesting Broker Architecture</td>
</tr>
<tr>
<td>CREATE</td>
<td>Computational Research and Engineering for Acquisition Tools and Environments</td>
</tr>
<tr>
<td>CWM</td>
<td>Common Warehouse Metamodel</td>
</tr>
<tr>
<td>dB</td>
<td>Decibel</td>
</tr>
<tr>
<td>DBMS</td>
<td>Database Management System</td>
</tr>
<tr>
<td>DAG</td>
<td>Defense Acquisition Guidebook</td>
</tr>
<tr>
<td>DARPA</td>
<td>Defense Advanced Research Project Agency</td>
</tr>
<tr>
<td>DAU</td>
<td>Defense Acquisition University</td>
</tr>
<tr>
<td>DCDR</td>
<td>Digital design from Critical Design Review (CDR)</td>
</tr>
<tr>
<td>DL</td>
<td>Descriptive Logic</td>
</tr>
<tr>
<td>DoD</td>
<td>Department of Defense</td>
</tr>
<tr>
<td>DoDAF</td>
<td>Department of Defense Architectural Framework</td>
</tr>
<tr>
<td>DoE</td>
<td>Design of Experiments</td>
</tr>
<tr>
<td>DSL</td>
<td>Domain Specific Languages</td>
</tr>
<tr>
<td>DSM</td>
<td>Domain Specific Modeling</td>
</tr>
<tr>
<td>DSMML</td>
<td>Domain Specific Modeling Language</td>
</tr>
<tr>
<td>E/DRAP</td>
<td>Engineering Data Requirements Agreement Plan</td>
</tr>
<tr>
<td>ERS</td>
<td>Engineered Resilient Systems</td>
</tr>
<tr>
<td>FAA</td>
<td>Federal Aviation Administration</td>
</tr>
<tr>
<td>FMEA</td>
<td>Failure Modes and Effects Analysis</td>
</tr>
<tr>
<td>FMI</td>
<td>Functional Mockup Interface</td>
</tr>
<tr>
<td>FMU</td>
<td>Functional Mockup Unit</td>
</tr>
<tr>
<td>Acronym</td>
<td>Full Form</td>
</tr>
<tr>
<td>---------</td>
<td>-----------</td>
</tr>
<tr>
<td>GAO</td>
<td>Government Accounting Office</td>
</tr>
<tr>
<td>HPC</td>
<td>High Performance Computing</td>
</tr>
<tr>
<td>HPCM</td>
<td>High Performance Computing Modernization</td>
</tr>
<tr>
<td>HW</td>
<td>Hardware</td>
</tr>
<tr>
<td>I&amp;I</td>
<td>Integration and Interoperability</td>
</tr>
<tr>
<td>IBM</td>
<td>International Business Machines</td>
</tr>
<tr>
<td>IBD</td>
<td>SysML Internal Block Diagram</td>
</tr>
<tr>
<td>ICD</td>
<td>Interface Control Document</td>
</tr>
<tr>
<td>ICTB</td>
<td>Integrated Capability Technical Baseline</td>
</tr>
<tr>
<td>IDEFO</td>
<td>Icam DEFinition for Function Modeling</td>
</tr>
<tr>
<td>IEEE</td>
<td>Institute of Electrical and Electronics Engineers</td>
</tr>
<tr>
<td>INCOSE</td>
<td>International Council on Systems Engineering</td>
</tr>
<tr>
<td>IPR</td>
<td>Integration Problem Report</td>
</tr>
<tr>
<td>IRL</td>
<td>Integration Readiness Level</td>
</tr>
<tr>
<td>ISEF</td>
<td>Integrated System Engineering Framework developed by Army's TARDEC</td>
</tr>
<tr>
<td>ISO</td>
<td>International Organization for Standardization</td>
</tr>
<tr>
<td>IT</td>
<td>Information Technology</td>
</tr>
<tr>
<td>IWC</td>
<td>Integrated Warfighter Capability</td>
</tr>
<tr>
<td>JCID</td>
<td>Joint Capabilities Integration and Development System</td>
</tr>
<tr>
<td>JEO</td>
<td>Jupiter Europa Orbiter project at NASA/JPL</td>
</tr>
<tr>
<td>JSF</td>
<td>Joint Strike Fighter</td>
</tr>
<tr>
<td>JPL</td>
<td>Jet Propulsion Laboratory of NASA</td>
</tr>
<tr>
<td>KPP</td>
<td>Key Performance Parameter</td>
</tr>
<tr>
<td>KSA</td>
<td>Key System Attributes</td>
</tr>
<tr>
<td>Linux</td>
<td>An operating system created by Linus Torvalds</td>
</tr>
<tr>
<td>LOC</td>
<td>Lines of Code</td>
</tr>
<tr>
<td>M&amp;S</td>
<td>Modeling and Simulation</td>
</tr>
<tr>
<td>MARTE</td>
<td>Modeling and Analysis of Real Time Embedded systems</td>
</tr>
<tr>
<td>MATRIXx</td>
<td>Product family for model-based control system design produced by National Instruments; Similar to Simulink</td>
</tr>
<tr>
<td>MBEE</td>
<td>Model-based Engineering Environment</td>
</tr>
<tr>
<td>MBSE</td>
<td>Model-based System Engineering</td>
</tr>
<tr>
<td>MBT</td>
<td>Model Based Testing</td>
</tr>
<tr>
<td>MC/DC</td>
<td>Modified Condition/Decision</td>
</tr>
<tr>
<td>MCE</td>
<td>Model-centric engineering</td>
</tr>
<tr>
<td>MDA®</td>
<td>Model Driven Architecture®</td>
</tr>
<tr>
<td>MDD™</td>
<td>Model Driven Development</td>
</tr>
<tr>
<td>MDE</td>
<td>Model Driven Engineering</td>
</tr>
<tr>
<td>MDSD</td>
<td>Model Driven Software Development</td>
</tr>
<tr>
<td>MDSE</td>
<td>Model Driven Software Engineering</td>
</tr>
<tr>
<td>MIC</td>
<td>Model Integrated Computing</td>
</tr>
<tr>
<td>MMMM</td>
<td>Modeling Maturity Model</td>
</tr>
<tr>
<td>MoDAF</td>
<td>United Kingdom Ministry of Defence Architectural Framework</td>
</tr>
<tr>
<td>MOE</td>
<td>Measure of Effectiveness</td>
</tr>
<tr>
<td>MOF</td>
<td>Meta Object Facility</td>
</tr>
<tr>
<td>MOP</td>
<td>Measure of Performance</td>
</tr>
<tr>
<td>MVS</td>
<td>Multiple Virtual Storage</td>
</tr>
<tr>
<td>NASA</td>
<td>National Aeronautics and Space Administration</td>
</tr>
<tr>
<td>NAVAIR</td>
<td>U.S. Navy Naval Air Systems Command</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NAVSEA</td>
<td>U.S. Naval Sea Systems Command</td>
</tr>
<tr>
<td>NDA</td>
<td>Non-disclosure Agreement</td>
</tr>
<tr>
<td>NDIA</td>
<td>National Defense Industrial Association</td>
</tr>
<tr>
<td>NEAR</td>
<td>Naval Enterprise Architecture Repository</td>
</tr>
<tr>
<td>NPS</td>
<td>Naval Postgraduate School</td>
</tr>
<tr>
<td>OCL</td>
<td>Object Constraint Language</td>
</tr>
<tr>
<td>OMG</td>
<td>Object Management Group</td>
</tr>
<tr>
<td>OO</td>
<td>Object oriented</td>
</tr>
<tr>
<td>OSD</td>
<td>Office of the Secretary of Defense</td>
</tr>
<tr>
<td>OSLC</td>
<td>Open Services for Lifecycle Collaboration</td>
</tr>
<tr>
<td>OV1</td>
<td>Operational View 1 – type of DoDAF diagram</td>
</tr>
<tr>
<td>OWL</td>
<td>Web Ontology Language</td>
</tr>
<tr>
<td>PDM</td>
<td>Product Data Management</td>
</tr>
<tr>
<td>PDR</td>
<td>Preliminary Design Review</td>
</tr>
<tr>
<td>PES</td>
<td>Physical Exchange Specification</td>
</tr>
<tr>
<td>PIA</td>
<td>Proprietary Information Agreement</td>
</tr>
<tr>
<td>PIM</td>
<td>Platform Independent Model</td>
</tr>
<tr>
<td>PLM</td>
<td>Product Lifecycle Management</td>
</tr>
<tr>
<td>POR</td>
<td>Program of Record</td>
</tr>
<tr>
<td>PRR</td>
<td>Production Readiness Review</td>
</tr>
<tr>
<td>PSM</td>
<td>Platform Specific Model</td>
</tr>
<tr>
<td>QMU</td>
<td>Quantification of Margins and Uncertainty</td>
</tr>
<tr>
<td>RT</td>
<td>Research Task</td>
</tr>
<tr>
<td>RFP</td>
<td>Request for Proposal</td>
</tr>
<tr>
<td>ROI</td>
<td>Return On Investment</td>
</tr>
<tr>
<td>SAVI</td>
<td>System Architecture Virtual Integration</td>
</tr>
<tr>
<td>SE</td>
<td>System Engineering</td>
</tr>
<tr>
<td>SERC</td>
<td>Systems Engineering Research Center</td>
</tr>
<tr>
<td>SETR</td>
<td>System Engineering Technical Review</td>
</tr>
<tr>
<td>Simulink/Stateflow</td>
<td>Product family for model-based control system produced by The Mathworks</td>
</tr>
<tr>
<td>SCR</td>
<td>Software Cost Reduction</td>
</tr>
<tr>
<td>SDD</td>
<td>Software Design Document</td>
</tr>
<tr>
<td>SE</td>
<td>System Engineering</td>
</tr>
<tr>
<td>SFR</td>
<td>System Functional Review</td>
</tr>
<tr>
<td>SLOC</td>
<td>Software Lines of Code</td>
</tr>
<tr>
<td>SME</td>
<td>Subject Matter Expert</td>
</tr>
<tr>
<td>SOAP</td>
<td>A protocol for exchanging XML-based messages – originally stood for Simple Object Access Protocol</td>
</tr>
<tr>
<td>SoS</td>
<td>System of Systems</td>
</tr>
<tr>
<td>Software Factory</td>
<td>Term used by Microsoft</td>
</tr>
<tr>
<td>SRR</td>
<td>System Requirements Review</td>
</tr>
<tr>
<td>SRS</td>
<td>Software Requirement Specification</td>
</tr>
<tr>
<td>STOVL</td>
<td>Short takeoff and vertical landing</td>
</tr>
<tr>
<td>SVR</td>
<td>System Verification Review</td>
</tr>
<tr>
<td>SW</td>
<td>Software</td>
</tr>
<tr>
<td>SysML</td>
<td>System Modeling Language</td>
</tr>
<tr>
<td>TARDEC</td>
<td>US Army Tank Automotive Research</td>
</tr>
<tr>
<td>TBD</td>
<td>To Be Determined</td>
</tr>
<tr>
<td>TRL</td>
<td>Technology Readiness Level</td>
</tr>
</tbody>
</table>
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

10 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 Vista™ 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.
11 References

[22] Blackburn, Mark, Robert Cloutier, Eirik Hole, and Gary Witus. Introducing Model-Based Systems Engineering


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


[70] IDEF®, Computer Systems Laboratory of the National Institute of Standards and Technology (NIST), 1993.


http://www.osti.gov/servlets/purl/976951-meC28s/

[105] Post, D., Computational Research Engineering Acquisition Tools and Environments, A DoD Program to Aid Acquisition Engineering, NDIA, October 2014.


Topper, S., Model Based Systems Engineering (MBSE), NDIA, 19-April-2016.


http://www.vitechcorp.com/products/core.shtml


