Contract No. HQ0034-13-D-0004

SYSTEMS ENGINEERING RESEARCH CENTER

Transforming Systems Engineering through Model-Centric Engineering

A013 Interim Technical Report SERC-2017-TR-111

August 8, 2018

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

Co-Principal Investigator: Dr. Dinesh Verma, Stevens Institute of Technology

Research Team

Georgetown University: Dr. Robin Dillon-Merrill

Stevens Institute of Technology: Roger Blake, Dr. Mary Bone, Brian Chell, Rick Dove, Dr. John Dzielski, Dr. Paul Grogan, Dr. Steven Hoffenson, Eirik Hole, Dr. Roger D. Jones, Dr. Benjamin Kruse, Dr. Kishore Pochiraju, Chris Snyder, Dr. Rob Cloutier

University of Massachusetts: Dr. Ian Grosse, Dr. Tom Hagedorn

Sponsor:

U.S. Army Armament Research, Development and Engineering Center (ARDEC), Office of the Deputy Assistant Secretary of Defense for Systems Engineering (ODASD(SE))

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

This material is based upon work supported, in whole or in part, by the U.S. Department of Defense through the Office of the Assistant Secretary of Defense for Research and Engineering (ASD(R&E)) under Contract HQ0034-13-D-0004 (TO 0068).

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

We wish to acknowledge the great support of the ARDEC and SERC sponsors for their support in Phase I and Phase II. We specifically want to thank the ARDEC leadership and leading coordinators of this SERC Research Task, including Jeff Dyer, Allan Lagasca, Eddie Bauer, Christina Jauregui, Cliff Marini and Matt Cilli, with new leadership for Phase II by Mitul Patel.

We sincerely apologize if we have missed anyone else that has supported our efforts.

Richard Swanson
Tony Farina
Luke Helsel
David Tam
Doug Ray
Phil Brislin
David Chau
Darryl Howell
Greg Nieradka
Roshan Patel
John Campbell
Jason Morales
June Ferrabolli
Mustafa Rawat
Chad Clark
Bill Miller
Adrian Celiz
Frank Salvatore
Sean Pham
Jamie Galiastro

RESEARCH TEAM

The following is a list of the researchers and their affiliations who contributed to Phase I and Phase II. Each researcher or contributor may be involved in one or more tasks that are now mapped to a linked set of research use cases. These relationships are described in greater detail in Section 2.

Affiliation	Researcher and Authors
Stevens Institute of Technology	Dr. Mark Blackburn (PI) Dr. Dinesh Verma (Co-PI) Ralph Giffin (Co-PI) Roger Blake Dr. Mary Bone Dr. Rob Cloutier (Visiting Professor) Rick Dove Dr. John Dzielski Dr. Paul Grogan Bob Hathaway Dr. Steven Hoffenson Eirik Hole Dr. Roger D. Jones Dr. Benjamin Kruse Myriam Marcus Dr. Kishore Pochiraju Chris Snyder Student researchers ▪ Brian Chell (Phase I & II) ▪ Khushali Dave (Phase I & II) ▪ Harsh Kevadia (Phase I) ▪ Kunal Batra (Phase I & II) ▪ Luigi Ballarinni (Phase II) ▪ Pasquale Montemarano (Phase I) Prior team researcher ▪ Andrew Dawson (Phase I) ▪ Dr. Deva Henry (Phase I) ▪ Jeff McDonald (Phase I) ▪ Dr. Gregg Vesonder (Phase I)

	▪ Dr. Lu Xiao (Phase
Georgetown University	Dr. Robin Dillon-Merrill
University of Massachusetts	Dr. Ian Grosse (Phase II) Dr. Tom Hagedorn (Phase II)
University of Southern California	Dr. Todd Richmond (Phase I) Edgar Evangelista (Phase I) Student and faculty researchers

EXECUTIVE SUMMARY

This research task (RT-168) addresses research needs defined by the United States (US) Army Research, Development and Engineering Command (RDECOM) Armament Research, Development and Engineering Center (ARDEC) in Picatinny, NJ. The purpose of this RT-168 Phase II final technical report is to document the refinement and expansion of those needs and the accomplishment provide through working sessions, demonstrations, presentations, models, prototype tools and reports provided to the ARDEC team, with particular focus on the updates since the start of Phase II in August 2017. These needs are characterized as overarching objectives and goals to elicit requirements for the Armament Virtual Collaboratory Environment (AVCE) integrated Model Based Environment (iMBE). The AVCE iMBE is ARDEC's envisioned concept of an integrated modeling environment - "the system for designing future ARDEC systems or systems-of-systems." The intent is to understand the relationships between Systems Engineering (SE) activities and methods in the context of a Digital Thread concept developed by ARDEC.

This research task focuses on the ARDEC-relevant needs for a transformation for systems engineering enabled by model-centric engineering (MCE). 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.

The Phase I research efforts created awareness about research challenges, opportunities and emerging trends. The efforts during Phase II have been adapted by the sponsor to focus on some of those research thrusts that contribute to the vision for the modeling and infrastructure for AVCE iMBE, such as approaches and technologies for integration and interoperability of multi-domain and multi-physics models, semantic web technologies, Multidisciplinary Design, Analysis and Optimization (MDAO), and system modeling. These technologies are often new, and the research also documents methods and lessons learned. Aligning with the leading-edge work from National Aeronautics and Space Administration (NASA) Jet Propulsion Laboratory (JPL) the team developed several Docker configurations for deployment of OpenMBEE that enables the use of the Model Development Kit/DocGen, the Model Management System (MMS) and View Editor. This instantiation of OpenMBEE has been integrated into our Integration and Interoperability Framework (IoIF) for our first use case to integrate SysML models with the ARDEC inspired Decision Framework, a decision ontology based on the Basic Formal Ontology using semantic

---

[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 RT48/118/141/157/170 for Model Centric Engineering (MCE).

web technology, with output visualizations using Tableau. ARDEC has identified about 80 tools that may be assembled into analysis workflows for any project, the integration of those tools for understanding cross-domain impacts is challenging. Therefore, our research thrusts are characterized by 16 related use cases look to demonstrate and evolve IoIF to research technological aspects that include cross-domain model integration, model integrity, ontologies, semantic web technologies, modeling methods, decision analysis framework, multi-physics modeling, and model visualization and integrated modeling environments supporting an authoritative source of truth (AST) that can contribute to AVCE iMBE.

During Phase II we continue to actively interact with our ARDEC sponsors and extend the Phase I engagements of five working sessions, one special session and 19 virtual meetings. For Phase II, we have produced the required deliverables such as this report, but also have conducted six additional working sessions and two special deep dive working sessions on ontologies and semantic web technology held at either Stevens Institute of Technology (Stevens) or Picatinny Arsenal. We have participated or led 10 virtual events, and presentations or demonstrations. We have contributed software to ARDEC from our IoIF, and development of a Docker installation for rapidly deploying the OpenMBEE. Finally, we are producing videos that capture demonstrations of our research and recommended methods for using different types of technologies such as OpenMBEE.

Finally, this research is being conducted in collaboration with three SERC research tasks sponsored by the Naval Air Systems Command (NAVAIR) under RT-170, RT-176 and RT-195, as well as Department of Defense (DoD) Digital Engineering (DE) Strategy released June 2018. Our research is also fostered by our relationships with NASA/JPL, the Open Collaboration Group for Model-Based Systems Engineering (MBSE) and Semantic Technologies for Systems Engineering Initiative.

PART I: RESEARCH TASK OVERVIEW

Part I provides and overview of this research task and sets the context for the needed research as defined and evolved by our sponsor, as well as the objectives, scope and organization of this report. This part also provides a summary of the current set of research use cases, our Phase II efforts, status, events, demonstrations, deliverables, models, prototype tools and recommendations based on our increased understanding of the research objectives.

1 INTRODUCTION

The SERC team has conducted eleven working sessions, four special sessions and 29 virtual meetings with the United States (US) Army RDECOM-ARDEC in Picatinny, NJ to discuss the needs and scenarios for a System Engineering (SE) transformation enabled by evolving model-centric engineering (MCE) technologies and methods since the start of this task in August 2016. This report blends updates from the start of Phase II into the evolving research results and accomplishments from Phase I. The results from Phase I led ARDEC to adapt early guidance into focused research that they believe to help advance their efforts on ARDEC's vision for an Armament Virtual Collaboratory Environment (AVCE) integrated Model Based Environment (iMBE).

We refined and expanded our research use cases to align with research thrusts discussed by our sponsor at the start of Phase II in September 2017. These include but are not limited to: cross-domain model integration, ontologies, semantic web technologies, modeling methods, decision analysis framework, multidisciplinary design analysis and optimization, model integrity, and model visualization and integrated modeling environments supporting an authoritative source of truth that can contribute to AVCE iMBE vision and system model.

We are also fostering bi-directional sharing of research interests and results with our US Navy Naval Air Command (NAVAIR) sponsors. We are collaborating in several MCE-related efforts to provide the opportunity to leverage and share with the Open Collaboration Group for MBSE and OpenMBEE [149], Semantic Technologies for Systems Engineering (ST4SE) initiative, DoD Digital Engineering Strategy [206], the Aerospace Industry Association (AIA) on Concept of Operations (CONOPS) for Government and Industry collaboration through MBSE [3] and the National Defense Industry Association (NDIA) Modeling and Simulation group who are coordinating working groups to investigate approaches for using Digital Models for competitive down select.

1.1 ARMAMENT VIRTUAL COLLABORATORY ENVIRONMENT VISION

The AVCE iMBE vision portrayed by ARDEC [10] reflects on their understanding of the research needed to advance to a future state of their integrated modeling environment. There are many enablers that relate to characteristics of a holistic approach that aligns with their vision such as (this list is not exhaustive, but represents advances in use today):

- Mission-level simulations that are being integrated with system of systems (SoS) and system simulation that increasingly interoperate with distributed interactive simulation capabilities, augmented virtual reality, and gaming technology

- Computer-aided Design (CAD), behavioral techniques, physics-based/engineering simulations, decision analytics, Computer-aided Manufacturing (CAM), system architecting, prototyping, embedded in a knowledge management environment
- Enabling collaborative environments by leveraging social media technologies and operational metaphors in an engineering context
- Multidisciplinary Design, Analysis and Optimization (MDAO) for trade study analyses through more systematic design of experiments allows engineers to make many more excursions through both the problem and the design spaces
- Engineering affordability analysis, which is a risk-based approach that could be used to significantly reduce physical tests by focusing on those system uses that have the most uncertainty about margins of performance
- Decision analysis framework
- Risk modeling and Bayesian-relevant analysis
- Platform-based approaches with virtual integration
- Pattern-based modeling based on ontologies with model transformation and analysis
- Domain-specific modeling languages
- Set-based design for more concurrent engineering and to keep design options open longer
- Modeling and simulation of manufacturing and possibly early prototyping
- Explosion of interactive visualization, which we will need as we have a “sea” of data and information derived from a “sea” of models with HPC computing capabilities

The updates at our February 2018 working session identified the current plan for ARDEC to continue focused efforts on AVCE iMBE, which include:

- Engineering and Analysis Workflow Development
- Workflow Analysis and Scope Determination
- Market Research
- Framework Technology Characterization/Assessment
- Identify Framework Alternatives
- Ontology / Semantic Web Research
- Systems Engineering Research Task

Our past, current and future research does map to these tasks [28]. The SERC’s research with NAVAIR Systems Engineering Transformation (SET) also provides considerable insights and research findings to support and extend the ARDEC research [26] [27][29].

The Deputy Assistant Secretary of Defense (DASD) has initiated a Digital Engineering Strategy [71]. ARDEC and NAVAIR are both participating in this initiative with the Digital Engineering Working Group. In addition, the SERC leadership confirmed and recommended that complementary research results can be shared across these research tasks. To the degree possible we are synergistically leveraging research completed or underway related to NAVAIR under SERC RT-157, RT-170, RT-195 and RT-176 that includes other research collaborators: Georgia Institute of Technology (Georgia Tech), University of Maryland, and the Naval Postgraduate School (NPS).

1.2 OBJECTIVES

The critical items gleaned from the ARDEC needs and our prior research resulted in the following set of proposed tasks:

- Task 1: Framework/architecture of development and collaboration environment that support cross-domain integration of models to address the heterogeneity of the various tools and environments
    - Accomplishments include IoIF integrating SysML, OpenMBEE, Decision Framework, Decision Ontologies, semantic web technologies, and visualization
- Task 2: Formalization of an information model for ARDEC-relevant domains to support capturing and sharing of data
    - Accomplishments include semantic web technologies and ontologies
- Task 3: Technology and domain-relevant modeling methodologies
    - Accomplishments documented in methods for MDAO, system modeling, CONOPS modeling
- Task 4: Demonstrations
    - Accomplishments demonstrated at various working session, with the most comprehensive of the IoIF integrating SysML, OpenMBEE, Decision Framework, Decision Ontologies, semantic web technologies, and visualization at working session #10
- Task 5: System Engineering Transformation Roadmap to roll out capabilities addressing all five perspectives in parallel:
    - Technologies and infrastructure
    - Methodologies and processes
    - People, training, competencies and framework viewpoints and interfaces
    - Operational & contractual paradigms for transformed interactions with industry
    - Governance

Earlier during Phase I, we decided to use research use cases that mapped to the ARDEC concept defined by their digital thread for AVCE iMBE. As reflected by abbreviated accomplishments listed against the tasks, the use cases cut across and overlap the tasks. The research continues to evolve to align with the needs of ARDEC. For example, Eddie Bauer's briefing for a Digital Engineering Working Group meeting stated: "*Research in Data Ontology/Information Model using semantic web ontologies is promising and could support model and simulation integration."* This led to some emphasis on specific use cases for Phase II, many that have been brought to bear in a demonstration IoIF integrating SysML, System Models, OpenMBEE, Decision Framework, Decision Ontologies, semantic web technologies, and visualizations with extensions to MDAO.

Some of the research results are emerging as elements of ARDEC's concept and architecture for AVCE iMBE as currently embodied in IoIF. There is understanding that semantic technologies provide potential to better understand the detailed information model in a semantically precise way and enables underlying computation capabilities to automate reasoning about systems engineering tasks. In addition, the semantic precision and cross-domain linkages of information enables more computational analytics about consistency, completeness and well-formed of captured information.

We are using a Model Based System Engineering (MBSE) approach to model our project, and also to assist ARDEC in assessing their AVCE iMBE models. We started to elaborate the research tasks using high-level use cases as shown in Figure 1, relating those use cases, and associating the use cases with the stakeholders involved in the research. The relationships between stakeholders and use cases reflects on the interactions and dependencies between the team's research.

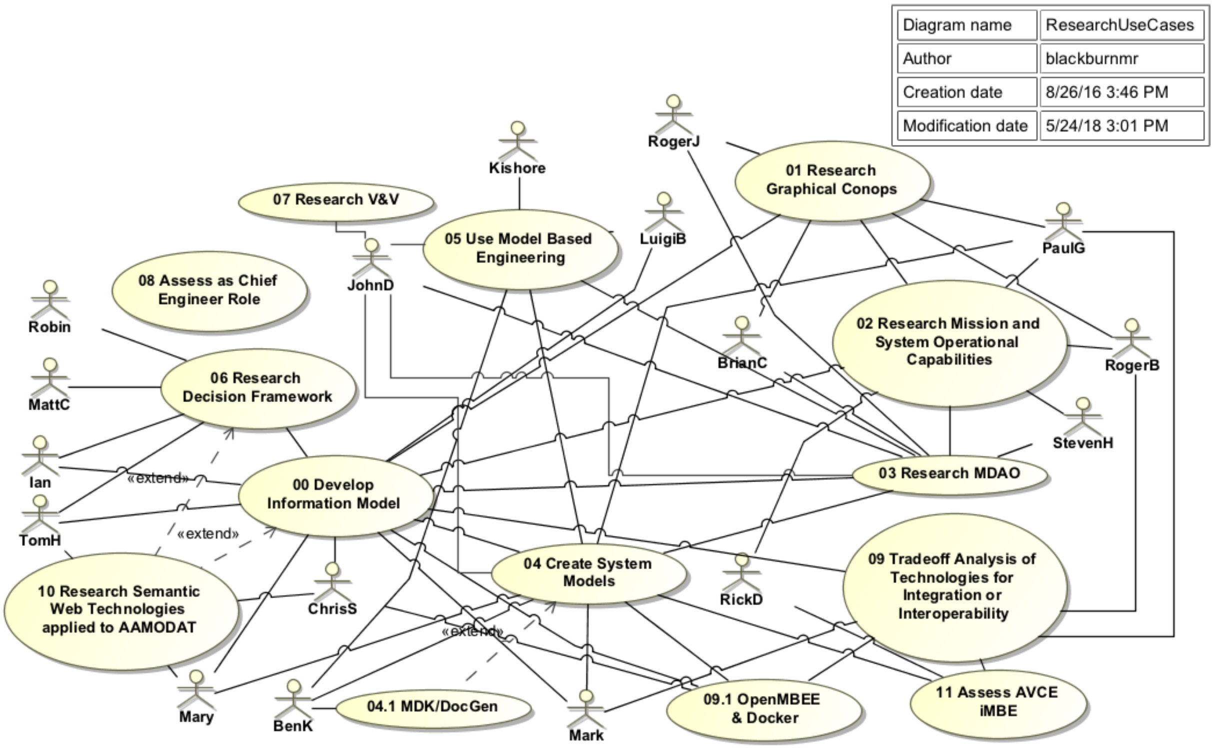


Figure 1. High-level Research Use Cases

Another perspective on the research thrust is shown in Figure 2. This reflects on the focus of semantic web technologies (SWT) and ontologies to support formalization of an underlying information model for cross-domain integration that is tool agnostic. SWT and ontologies formalize systems engineering knowledge and can enable computational reasoning to support enforcements of modeling methods that can ensure models and simulation are used in a way that leads to trust in the predictions from those models (i.e., model integrity). MDAO is another way to use parametric approaches cutting across domains for analyzing trades at the mission, system and subsystem levels. All of these technologies provide insights into the needs of an integrated modeling environment. We summarize and organize the research in a manner used on RT-168 as use cases (UC) that cut across the evolving case studies as it relates to Figure 2.

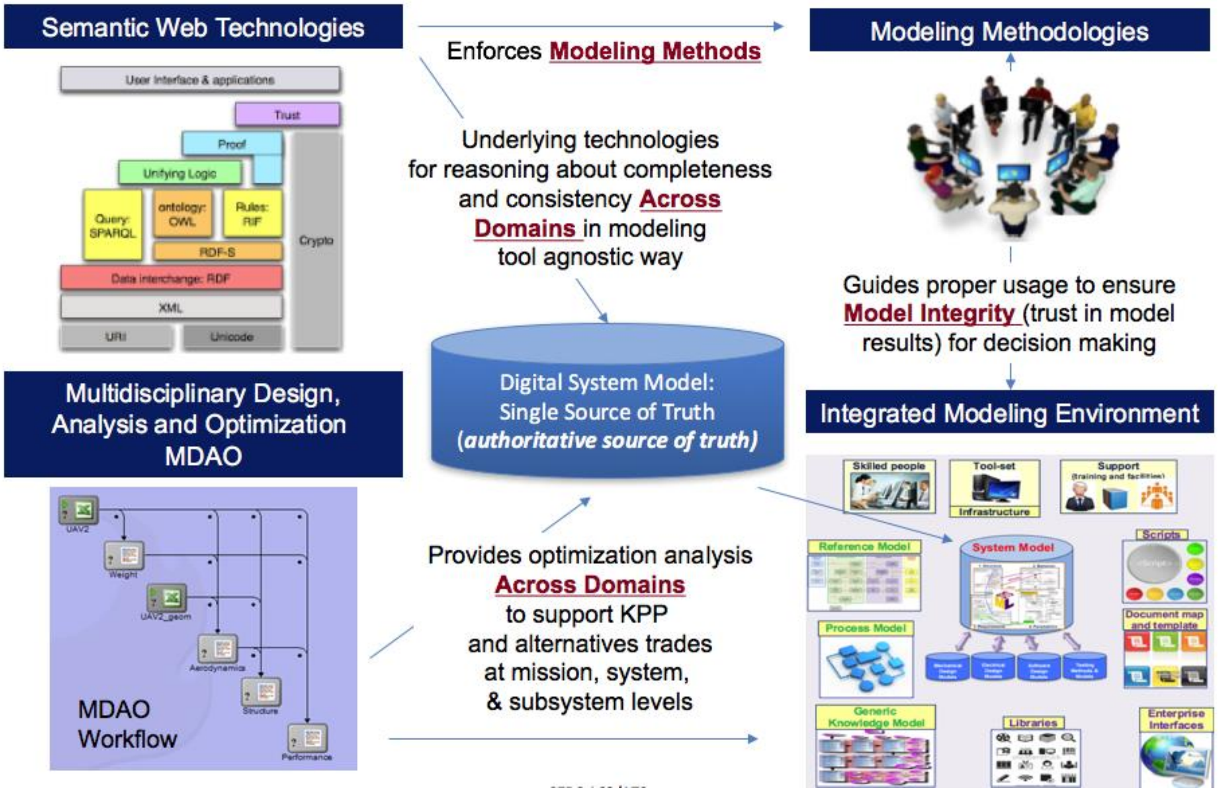


Figure 2. Cross-cutting Relationships of Research Needs

1.3 SCOPE

Early in this research, we characterized a simplified perspective on “traditional systems engineering” process phases and views to discuss how models that produce information for these views provide a basis for an underlying information model that captures information to enable a Decision Framework based on the Integrated Systems Engineering Decision Management (ISEDM) process [49][50], as shown in Figure 3. We notionally define:

- Concept of Operations (CONOPS) derived from simulation and gaming technologies
- “What” we want – requirements and constraints
- “How” (1 or more) – designs to achieve the “What”
- “How well” (usually many) to assess the “How” using analysis, testing, reviews and assessing how the design satisfies the requirements, given the constraints to achieve the mission concept
- The underlying Information Model links the data or metadata from many different domains
- The Decision Framework, we believe can demonstrate how data from the information model can be used to populate the Decision Framework in the form of the implementation of AAMODAT with potential refinements and extensions supporting a method to determine the Key Performance Parameters (KPPs) of the various stakeholders.

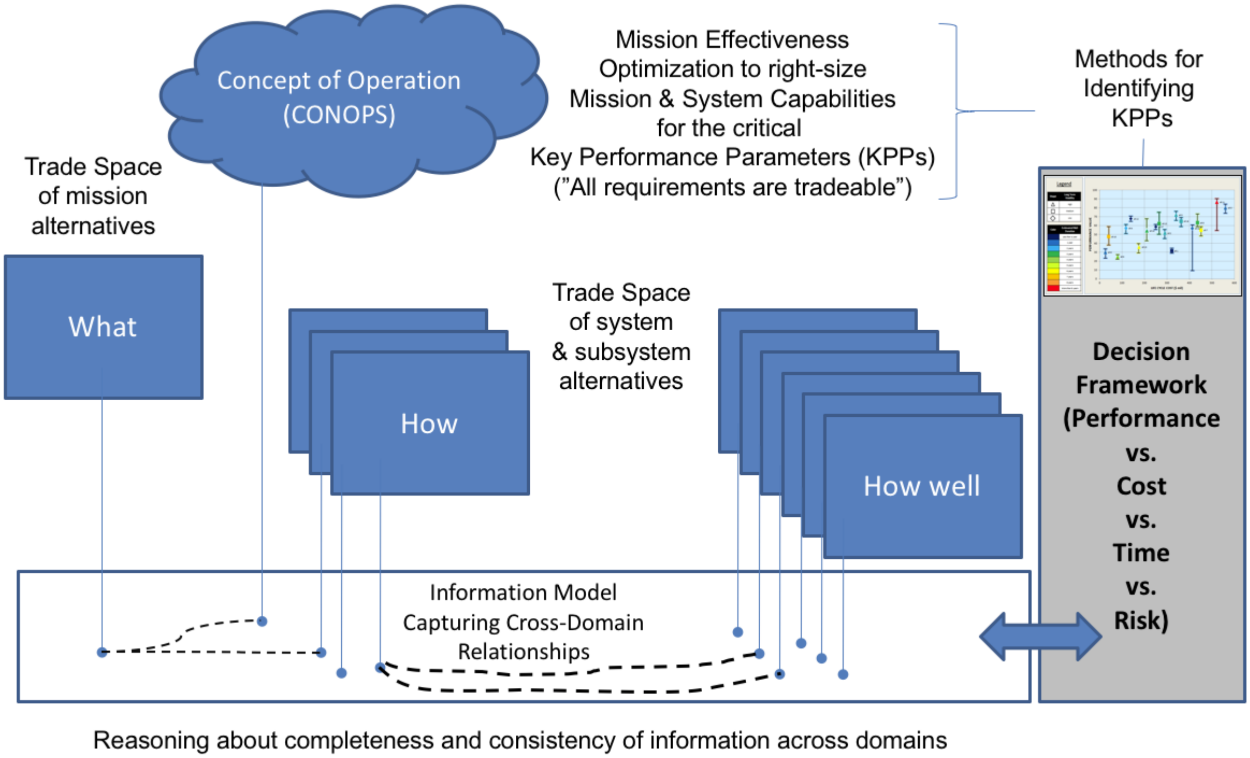


Figure 3. Context of System Engineering of Challenge Areas

MCE is enabled by computational technologies which now provide a means for using modeling and simulation in a transformed approach to systems engineering. A key problem is that most of these technologies are not integrated currently (and many may never be). ARDEC confirmed this issue during Phase I, at a January 2017 working session. This was further acknowledged in various talks at the NASA/JPL Symposium and Workshop on Model Based System Engineering held January 25-27, 2017. Therefore, we are interested in approaches that leverages tool-to-tool integrations where feasible (e.g., MagicDraw and ModelCenter through MBSEPak), but the research is targeted on approaches to using data interoperability as a means for accomplishing integration, when tool-to-tool integration is not feasible or not cost-effective. We plan to do research in the other two areas of Mission and Systems and understand the flow of information needed to be linked between them, then characterize those linkages in an Information Model. Our research accomplishments in this area includes the development of an evolving IoIF, which has been demonstrated to ARDEC, and we have provided the various models and software for evolving versions of IoIF.

ARDEC's Dr. Matt Cilli during Phase I believed that information can be captured to drive the Decision Support Model Construct [49] (referred to as Decision Framework). During Phase II we formalized information to demonstrate the feasibility of this concept, now integrated in IoIF. The Decision Framework with a tool implementation serves many purposes and benefits:

- Provides senior management and program managers with visual representations of key tradeoff defined in terms of KPPs such as Performance, Cost, Time, and Risk
- As shown in Figure 4, scatterplot shows in a single chart how system level alternatives respond in multiple dimensions of stakeholder value
- Assessment Flow Diagrams (AFDs) trace the relationships between physical means, intermediate measures, and fundamental objectives
  - This has been formalized during Phase II using SysML, MBSEPak, and ModelCenter

- Provides methodological guidance for identifying KPPs
- Can be used with uncertainty analysis as a measure for understanding maturing design
- Enables bi-directional analysis throughout lifecycle

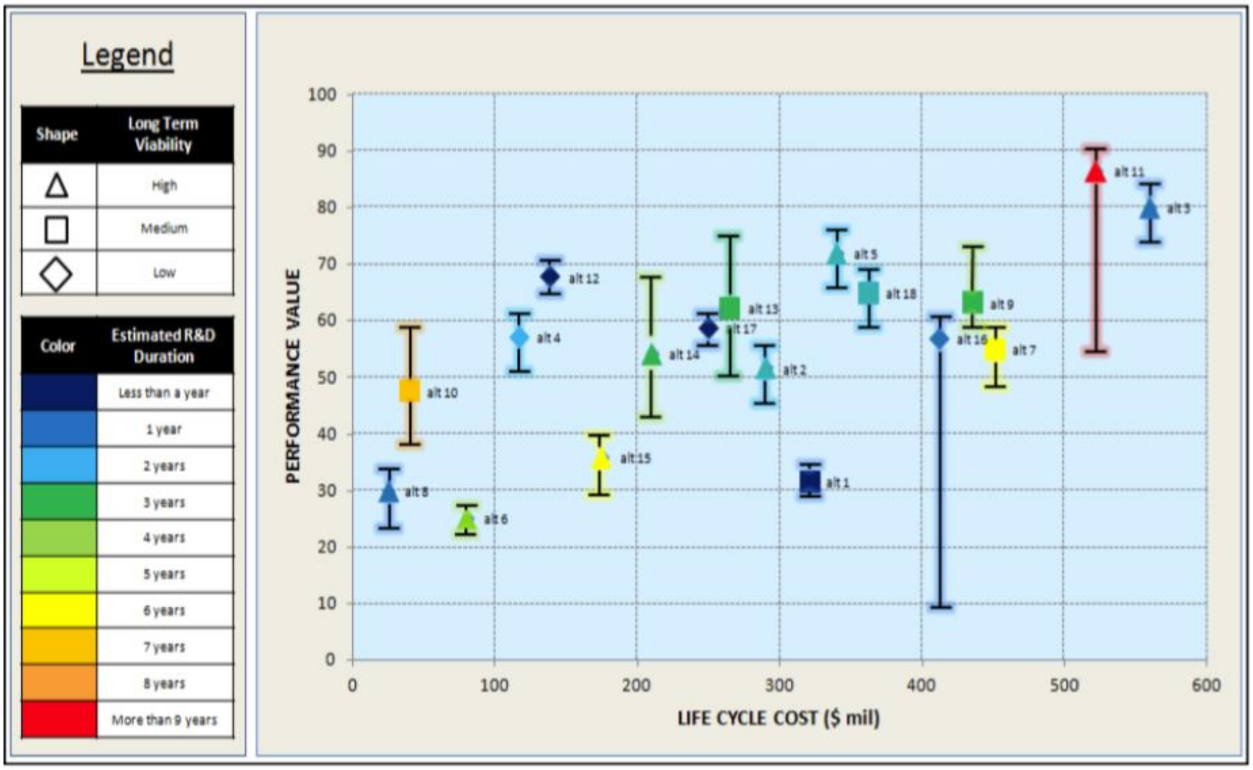


Figure 4. Decision Support Model Construct

The team has demonstrated the creation and use of SWT and ontologies, both standalone and as part of IoIF, and is working directly with ARDEC to develop ontologies relevant to ARDEC. We have created a decision ontology to provide support for the Decision Framework concept discussed in Figure 2, also demonstrated as part of IoIF. We have made significant strides during the past year in formalizing the Decision Framework elements, and demonstrated this concept at several events, including public events such as the Phoenix Integration International Users' Conference held in Annapolis, Maryland, on April 17 – 19, 2018 [23]. Our sponsor has stated that they believe the efforts to date have helped ARDEC in making decisions on approaches to the development of requirements and architectures for AVCE iMBE.

The team has constructed several variants of artificial unmanned aircraft systems (UAS) scenarios (use cases) and evolving scenarios that demonstrate methods to address many of the cross-cutting concerns from CONOPS, mission and system engineering. Mission-level scenarios have been created and demonstrated using four different modeling and simulation capabilities ranging from low-cost and low-fidelity to high-cost and high-fidelity. We have numerous examples for using MDAO at the CONOPS, mission and system level.

We have obtained and use academic licenses for some of the most powerful commercial tools in order to address research questions in the context of these types of tools; these are the types of tools used by both ARDEC and industry. This approach also addresses some organizational and domain-specific concerns. Through digital means we can now also encode historical knowledge in reference models, model patterns to embed methodological guidance to support continuous orchestration of analysis through new modeling metrics, and automated workflow to accelerate concepts to prototypes, deployment, and foster event-driven collaboration. Therefore, the deliverables include reports, demonstrations, meetings, meeting notes, Docker installers for OpenMBEE, software for IoIF, and examples of models developed so not to violate any of the academic licensing guidelines.

1.4 COLLABORATIVE RESEARCH SYNERGIES

Finally, ARDEC is also involved in synergistic collaborative efforts with NAVAIR and the Digital Engineering (DE) Working Group led by the Office of Deputy Assistant Secretary of Defense for Systems Engineering (ODASD(SE)), and we are working to align the research, to the extent possible, with the five DE Transformation goals [71] [206] that include:

- G1. Formalize the development, integration and use of models to inform enterprise and program decision making.
- G2. Provide an enduring authoritative source of truth.
- G3. Incorporate technological innovation to link digital models of the actual system with the physical system in the real world.
- G4. Establish a supporting infrastructure and environment to perform activities, collaborate and communicate across stakeholders.
- G5. Transform a culture and workforce that adopts and supports Digital Engineering across the lifecycle.

As is reflected in Figure 5, many of the research topics under investigation by ARDEC align with the DE Transformation goals. In addition, the mapping in Figure 5 shows that the research areas have significant overlap with some of the DE goals. This means that in order to achieve some of the goals, it will be necessary to have successful research outcomes across many research areas. Therefore, in the Part II of this report, the research areas are defined as cross-cutting use cases rather than specific tasks.

Future Research Areas	**G1.** Formalize the development, integration and use of models to inform enterprise and program decision making.	**G2.** Provide an enduring authoritative source of truth.	**G3.** Incorporate technological innovation to link digital models of the actual system with the physical system in the real world.	**G4.** Establish a supporting infrastructure and environment to perform activities, collaborate and communicate across stakeholders.	**G5.** Transform a culture and workforce that adopts and supports DE across the lifecycle.
Cross-discipline integration of models to address the heterogeneity of the various tools and environments using semantic technology	**X**	**X**	**X**	**X**	**X**
High Performance Computing (HPC) advancements such as; 1) supporting organizing and analyzing “Big Data” and 2) being able to program in parallel to take advantage of HPC capabilities, are needed to support the DE effort	**X**	**X**	**X**	**X**	
Model integrity to ensure trust in the model predictions by understanding and quantifying margins and uncertainty	**X**	**X**	**X**	**X**	**X**
Modeling methodologies that can embed demonstrated best practices and provide computational technologies for real-time training within digital engineering environments	**X**		**X**	**X**	**X**
Model composability to understand the possibilities, constraints and rulesets for composition of multiple models	**X**		**X**		
Human-model task allocation to understand what activities are best performed by human decision makers and what can effectively be automated or augmented with model intelligence					**X**
Workforce development to understand what is needed to educate model developers, users and decision makers to work in a DE environment					**X**
MCE acquisition to understand the needed changes to acquisition and security when developing in the new DE environment	**X**	**X**		**X**	**X**

Figure 5. Future Research Areas Mapped to Goals of Digital Engineering Transformation Strategy

These use cases will also investigate continuing synergistic research with the NAVAIR under RT-195, Semantic Technologies for Systems Engineering (ST4SE) initiative, RT-176 and other potential SERC research that is aligned with the principles and concepts for the SET as well as the ODASD(SE) Digital Engineering Strategy [71] [206].

1.5 ORGANIZATION OF DOCUMENT

Part I provides an overview of the research task.

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

Section 2 provides a summary of the current set of research use cases, our Phase II efforts, status, events, demonstrations, deliverables, and recommendations based on our increased understanding of the research objectives.

Part II describes the details for each research Use Cases (UC) and other collaborative research efforts.

Section 3 discusses the concept of the Information Model (UC00) underlying the AVCE, including ontologies and semantic web technologies (SWT); the fundamental purpose is to provide a means to link information and metadata from disparate sources across the various domains. This use case enabled by IoIF (UC09) provides a type of integration of several of the use cases.

Section 4 describes the concept for researching the use of Graphical CONOPS (UC01), including the potential relationships with MDAO (UC03).

Section 5 describes research into the use of mission and system modeling and simulation (UC02), and its relationships to graphical CONOPS and MDAO (UC03).

Section 6 discusses methods and examples using MDAO (UC03).

Section 7 systems modeling methodologies and MBSE (UC04), including details on associated set of tools based on OpenMBEE, the Model Management System (MMS), View Editor, and Model Development Kit (MDK)/DocGen.

Section 8 provides an overview of the approach for relating system models using MBSE, Model Based Engineering (MBE) (UC05), but more importantly for understanding the ways to link MBE models through the MCE toolchain as it relates to requirements for AVCE. Some of the details of the Courter UAS are covered in this use case, and a new section on Automated Concurrent Engineering.

Section 9 discusses the Decision Framework (UC06) research approach to leverage information captured through all of the phases and types of modeling into the information model to systematically populate SWT based on a decision ontology (UC00) and prototyped in IoIF with OpenMBEE (UC09).

Section 10 discusses how to use MCE for V&V (UC07), and the specific use of Monterey Phoenix (MP) for V&V of requirements, where we have graphically formalized the MP language using SysML activity diagrams and then transform the graphics into the MP language for automated V&V.

Section 11 is a use case to develop and assess the operational elements of the entire framework in the context of a Chief Engineer Role (UC08), where we have used our research for a Stevens Institute of Technology course on Cyber Physical Systems.

Section 12 describe tradeoff analysis of technologies for integration or interoperability (UC09) as a way for representing and analyzing the architecture trades for the requirements of AVCE. In addition, this section reflects on some of the most advanced integrated modeling environment identified through the NAVAIR related SERC research tasks, and our development of IoIF in the context of OpenMBEE.

Section 13 discusses the use of Semantic Web Technologies applied to Integrated Systems Engineering Decision Management (ISEDM) process [49], which has been generically referred to as the Decision Framework (UC06), and is now integrated with a decision ontology as part of IoIF (UC09).

Section 14 provides a summary based on assessing the AVCE iMBE requirements and model (UC11).

Section 15 provides a description of some of the SERC research synergies that are relevant to the ARDEC research objectives.

Section 16 provides a summary of Part II.

PART III: Appendices of Research Details. This appendix provides some additional details provided at the request of our sponsor and provides additional details about research details provided to our sponsor.

2 IN-PROCESS SUMMARY

This section provides a summary of the use cases shown in Figure 1 and shows those researchers involved in each use case. A summary of the working sessions, special sessions, events and deliverables provided to ARDEC is provided in Appendix 19J.

2.1 USE CASE SUMMARY

This section provides a high-level summary of each use case and recent results. Part II of this report (starting with Section 3) provides additional details on each use case (UC). As shown in Figure 1, there is considerable emphasis on understanding many of the cross-domain dependencies and relationships of the research use cases, and understanding the methods that must be used to guide the production of information, mostly in the form of models, across the various domains and lifecycle phases. There is also increased capability to examine the alternative analyses at the CONOPS level as demonstrated by using MDAO (UC03) to wrap graphical CONOPS (UC01) that has the potential to begin to examine tradespace analyses between mission and system levels.

We continue to evolve the IoIF as part of UC09 and integrate other capabilities with emphasis of demonstrating interoperability through SWT. For example, as shown in Figure 6, we have demonstrated the Decision Framework (UC06) enabled by SWT (UC00) with a decision ontology starting from a system model in SysML (UC04). This system model represents a number of UAV alternatives derived from a book chapter developed by Matt Cilli [50]. We leverage and demonstrate tool-to-tool integrations, for example the UAV SysML model integrates with ModelCenter, through MBSEPak, to illustrate the MDAO concept (UC03) for alternative analysis. The demonstration uses OpenMBEE MDK plugin to transfer SysML information to MMS. Recently developed IoIF capabilities transform the SysML information stored in MMS into the IoIF SWT to align with the decision ontology. We have created several instantiations of OpenMBEE [150] with a Docker [72] configuration, which is an underlying element of IoIF including MMS and View Editor. The transformed information from MMS, now stored in IoIF SWT is transformed into a representation to support visualizations of the various tradeoffs in Tableau [187]. IoIF now provides a substantial foundation for follow-on research and other synergies that have been discussed with our sponsor about elevating the Decision Framework concept in the context of IoIF to mission scenarios, or combinations of mission scenarios given system capabilities that can be composed into mission capabilities.

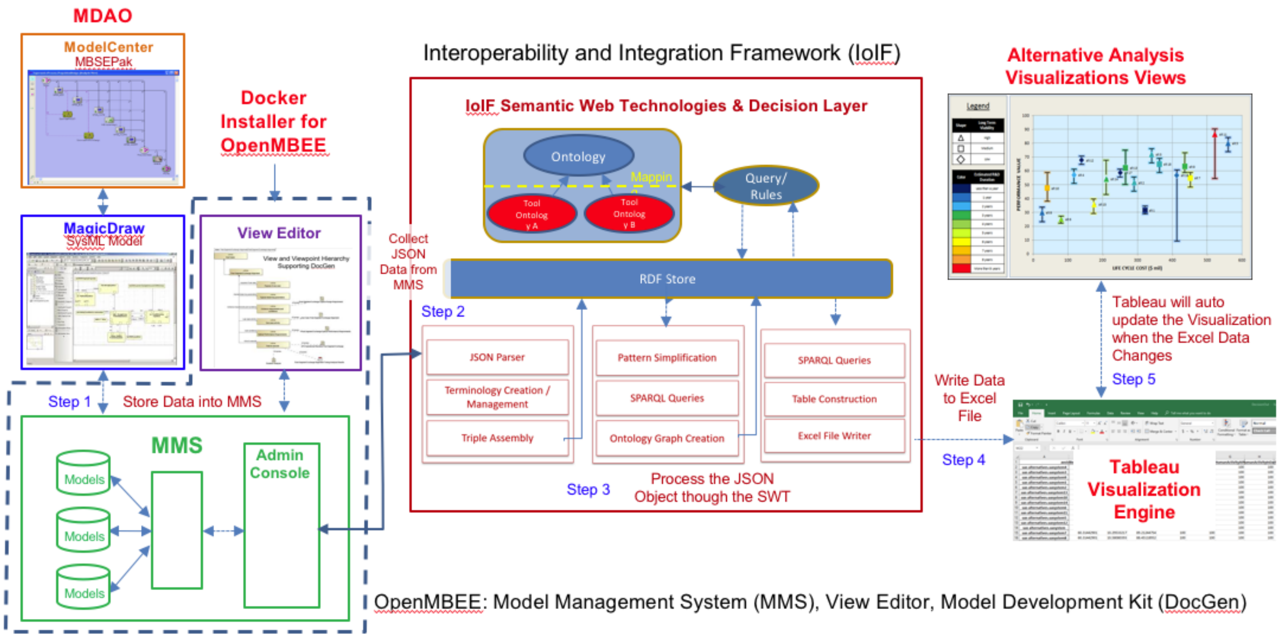


Figure 6. Interoperability and Integrating Framework (IoIF)

As part of the research to demonstrate the operational concepts of an Authoritative Source of Truth (AST) [205] [29], we have a working version of OpenMBEE and other tools hosted on an Amazon Web Service (AWS). We have made these models and information available for viewing our efforts on the on the NAVAIR System Engineering Transformation Surrogate Pilot project [29]. In collaboration with NAVAIR and NASA/JPL, we have started to use Integrated Model Centric Engineering (IMCE) ontologies [103] for systems engineering as part of the NAVAIR surrogate pilot in conjunction with our collaboration with the Semantic Technologies for Systems Engineering (ST4SE) initiative.

The following provides a brief summary of the research use cases, and for each use case discusses some of the different relationships to other use cases that are reflected in Figure 1.

00. Develop Information Model. This information model characterizes the underlying information and relationships to information and models that might need to be produced by the tools of AVCE, although we are using tools available to our Stevens laboratory. We are using the SWT language Web Ontology Language (OWL) [199] as the primary means for characterizing the information model across many of the use cases. As reflected in Figure 3, the challenge is to characterize this information for each of the various domains, including requirements, risks, designs (e.g., electrical, mechanical, etc.), and analyses. This reflects why there are so many associations from the other use cases. In addition, we have shown that it is technically feasible to capture this information and provide it as input to the Decision Framework (UC06) using SysML models and the Phoenix Integration MBSEPak and ModelCenter [23]. Recent accomplishments include:

    - Several demonstrations illustrating the feasibility of this concept in both working sessions and webinar sessions.
    - The SWT is architecturally represented in the IoIF as part of UC09, which was also demonstrated.

- Several deep-dive working sessions have focused on various type of ontology and SWT technology examples, underlying fundamentals and languages, and hands on use of Protégé, which is an ontology modeling tool and using a triple store (e.g., RDF4J).

01. Research Graphical CONOPS. Investigate the use of Graphical CONOPS technologies such as gaming environments. The team has created demonstrations using the Unity gaming engine [190] for simulating two autonomous UAS interacting in an environment. Our research collaborators USC/ICT have been evolving a technology called Early Synthetic Prototyping (ESP). We are fundamentally interested understanding if there is an underlying metamodel of the information that can be captured, regardless of the domain, and the methods that would be used to ensure that information is fully captured. This information would be mapped to the Information Model (UC00) and be provided as input to UC02. In addition, we have demonstrated how the parameters of simulation entities can be used in MDAO (UC03) [23].

- Newest use case investigates an extension of the prior work to using the Graphical CONOPS technologies Unity gaming engine with MDAO (UC03) using ModelCenter. The team has extended the prior demonstration extending the initial autonomous UAS capability
- There have been more than 10 updates to the Graphical CONOPS, which provides two types of missions for red/blue surveillance missions for autonomous quadcopters. The updated simulations include more realistic battery and flight models (UC05), and current research is using MDAO (UC03) for this level of the mission analysis.
- Modification to the prototype in order to run simulations when wrapped using MDAO faster than real-time

02. Research Mission and System Operational Capabilities. Investigate the methodological and relevant technologies for mapping the Graphical CONOPS into Mission and System modeling and simulation capabilities. The current research involves the use of VT MAK [115] and other 2D modeling and simulation environments for distributed simulations. We envision that information from UC01 would provide parameter information that can be refined or expanded. Therefore, like UC01, we want to understand the underlying information (e.g., metamodel) that would be mapped to the Information Model (UC00), and the associated methods for how to develop models at this level. This use case is also researching the relationships of these simulation models and system models in languages such as SysML.

- We have created a simple ontology as the basis to demonstrate information sharing through SWT to illustrate transfer of information through the SWT components of the IoIF. The demonstration also illustrated the use of triple stores and SPARQL [201] queries to store, extract or transform data in the SWT. The next planned demonstration will use these IoIF capabilities to transfer data between the Graphical CONOPS simulation and low fidelity mission-level analysis on a 2D plane with spatial positions of entities.
- This use case is also researching the relationships of these simulation models and system models in languages such as SysML
- Dr. Peter (Pete) Korfiatis attended the July 31, 2017 special session and recommended that Omar Valverde from MITRE provide an overview and demonstration of their 'Graphical CONOPS “integrated” with Rhapsody-driven (System Model) Simulations.' Like the use case UC01, they have used the Unity gaming engine and integrated it with a number of tools that we would categorize in UC04 (e.g., MBSE – Rhapsody/SysML) and UC05 (e.g., MBE, Modelica, MATLAB), including human inputs to test concepts in real-time

03. Research MDAO. Investigate the methods to trace capabilities to the relevant design disciplines and perform cross-domain analyses through MDAO for problem and design tradespace analyses. In addition, to characterizing elements of the framework, cross-domain relationships, but also characterize the methods used to support MDAO in a tool independent manner (we obtained academic licenses for ModelCenter, because we know that ARDEC uses that tool; these licenses can be used to provide examples, but not contribute to any ARDEC-specific work).

- Recent updates of unmanned aerial vehicle (UAV) model using MDAO workflows in ModelCenter show more realistic results in terms of weight and size, including use of Computational Fluid Dynamics (CFD), Finite Element Analysis (FEA), results of Design of Experiments (DOE) for range vs. cruise altitude vs wingspan, and a Pareto frontier for range, payload, and endurance as KPPs, new visualizations provided by version 12 of ModelCenter
- Another model that used Phoenix Integration MBSE Analyzer to integrate MagicDraw SysML with ModelCenter
- Demonstrated use of ModelCenter/MDAO to the Graphical CONOPS (UC01)
- Formalization of the Decision Framework Assessment Flow Diagram (UC06) using SysML, MBSEPak and ModelCenter
- See Applications for Three Research Use Cases in Model Centric Engineering using ModelCenter and MBSEPak, Phoenix Integration Webinar, Feb 7, 2018, https://www.phoenix-int.com/learn-more/webinars/

04. Create System Models. This applies MBSE to the case study examples and looks at how metamodels or metadata is represented in the Information Model (UC00) to provide traceability through the other forms of modeling for UC01, UC02, UC03 and UC05. This use case is developing different variants of UAS system models at both the system and mission level.

- Demonstrations include the use of the OpenMBEE Model Development Kit (MDK) DocGen to a number of models including the AVCE iMBE and Rotocopter UAV
- Created UAV SysML models
- We have an evolving SysML model for the RT-168 IoIF framework (UC09) to formalize the architecture, which has been provided to ARDEC
- Demonstrations and videos of the OpenMBEE environment, including the Model Management System (MMS) and View Editor components that have been open-sourced by NASA/JPL at: http://www.openmbee.org/ this is planned to be integrated with the IoIF framework
- Created representation is SysML for Monterey Phoenix (MP) to demonstrate the potential to perform early V&V requirements and architecture models [81]. Currently, MP is a language, but we believe we can develop a graphical language using SysML activity diagram (maybe profiled), and then use DocGen to extract information in order to translate into MP. This tasks benefits ARDEC, because RT-176 is funded by NAVAIR.
- Surrogate pilot for the NAVAIR SE Transformation
    - SysML models for the Surrogate pilot project models, including using the NASA/JPL system profiles for the ontologies
    - SysML models for the mission models
    - SysML for the systems models
    - Statement of Work models for source selection
    - Issue tracking model

    - Modularization method to partition the various types of models
- Investigating model management methods for various combination of OpenMBEE/MMS and No Magic Teamwork Cloud

05. Use Model Based Engineering. This applies Model-Based Engineering (MBE) typically associated with the different design disciplines (e.g., electrical, mechanical, controls) and will focus on some related research associated with counter UAS. Like UC04, we are interested at how metamodels from these various domain or metadata are represented in the Information Model (UC00) to provide traceability. It is currently acknowledged that, except for a few exceptions there is a gap in mapping from these types of modeling technologies to MBSE models.

    - Presented a session on “Representation Methods, Model Frameworks and Verification Tools for CPS Design” for UAS
    - Presentation on: Architecture and Prototyping of System Simulation with Semantic Data Exchange, which investigates bringing MBE design information into the SWT using an architecture and prototyping of system simulation with semantic data exchange; this will look at discipline-specific ontologies for cross-domain integration [35]

06. Research Decision Framework. As discussed in Section 1.3, we have had discussions with the ARDEC leads, who are intimately familiar with this framework and the evolving tool called AAMODAT. Fundamentally, a key goal for UC00 is to capture information that can be used to provide input to the Decision Framework (UC06). This would provide senior leaders and program managers the type of information they need to consider technology capability tradeoff using Performance, Cost (Affordability), Time (delivery schedule) and Risk. Fundamentally, if a particular answer was unacceptable, using the concept discuss herein, we could trace linkages through the Information model back to all other related perspectives on the system (UC01, UC02, UC03, UC04, UC05).

    - A major accomplishment during Phase II is the formalization of representations of Assessment Flow Diagram (AFD) using SysML and MDAO workflows, then now integrates through OpenMBEE MMS to a decision ontology, with SWT queries to produce Tableau visualizations as an alternative to AMODAT, packaged as part of IoIF.
    - We are also leveraging prior research from our University of Massachusetts collaborators that are evolving a decision ontology to be integrated into IoIF for capturing information related to the Decision Framework.
    - We provided demonstrations using SWT to get example data from DBpedia (which is a crowd-source effort to extract structured information from Wikipedia and make this information available on the Web) of a simple aircraft ontology and properties to show semantically rich data extracted from DBpedia using SWT tools (Protégé, OWL Viz, RDF).
    - Working on templates for different type of objective hierarchies (e.g., portfolio, product); objective hierarchies map into Key Performance Parameter (aka Key Parameters) in the AFD [49][50].

07. Research Verification and Validation (V&V). This use case was not considered in the original plan, but MCE does provide some unique opportunities to be more effective at contributing V&V evidence in early design. Rigorously defined models can directly support V&V, and this could both subsume cost and risks. This use case can likely identify candidate requirements for AVCE.

- As discussed in UC04, we are trying to leverage work with the SERC RT-176 led by Kristin Giammarco to use Monterey Phoenix (MP) for demonstrating the potential to perform early V&V requirements and architecture models [81], and have a SysML modeling concept based on activity diagrams that was developed during Phase II.

08. Assess as Chief Engineer Role. This use case is created so that one of our researchers, experienced in actual systems engineering can provide some level of assessment of our overarching approach and contribute to the requirements for AVCE. We too want to bring as many technologies as possible into our lab at Stevens in order to assess the gaps, but are also interesting in bring in master's students to using methods derived from this research.

- Our first experience is in using the OpenMBEE Model Development Kit/DocGen for the Stevens SYS-673 course – Implementing Cyber Physical System: Bringing Solutions to Life
  - The projects generated with DocGen are available on our Amazon Web Service that hosts OpenMBEE.
- We are also using several of these capabilities in experiments as part of the NAVAIR surrogate pilot; scenarios of the progress of the surrogate pilot for the NAVAIR Systems Engineering Transformation are on All Partners Access Network (APAN) apan.org [6].

09. Tradeoff Analysis of Technologies for Integration or Interoperability. This use case has been renamed and expanded due to information learned about other technologies that provide a means for looking at alternative technologies and approach to support either tool integration or some type of equivalent interoperability approaches that can be used for AVCE. Specifically, we continue to learn about the technologies and tools used by ARDEC and used in the case study to focus this research, as our interactions with different subject matter experts expands. In addition, this tasks revisits some of the most advanced tool integrations that have been developed by NASA/JPL [67] [11], the DARPA META projects [9] [8], Engineered Resilient Systems [93], Airbus [88], and generalization of commercial and industry integrated modeling environments. We assessed Windchill as part of this use case. We learned about Syndeia by Intercax, and coordinated a demonstration with our ARDEC sponsor. We are part of the leadership team for the Open Collaboration Group for MBSE and OpenMBEE [150].

- As discussed at the beginning of this section, the IoIF as shown in Figure 6, brings a number of use cases together:
  - The SWT is being expanded to support interoperability from Graphical CONOPS (UC01) to Mission-level simulation (UC02)
  - We are modeling this architectural framework (UC04)
  - We demonstrated using SWT in IoIF component (UC05)
  - We are formalizing a decision ontology (UC00) and data extraction mechanisms to store as much information as possible to support replacing the computational mechanisms of AAMODAT (UC10) directly within a triple store with SWT technologies to support the Decision Framework (UC06)
  - We have several instantiations of OpenMBEE to investigate using IoIF as a distributed communication mechanism to construct an implementation for an Authoritative Source of Truth (AST)
- Code for IoIF has been provided to ARDEC
- We have an operational Docker that does work with both OpenMBEE MMS and the View Editor

10. We are formalizing a decision ontology (UC00) and data extraction mechanisms to store as much information as possible to support replacing the computational mechanisms of AAMODAT (UC10) directly within a triple store with SWT technologies to support the Decision Framework (UC06)
    - We discussed how the use of the Decision Framework with AAMODAT is usually something that happens early on for ARDEC, and all over the project. It has helped to identify Key Performance Parameters (KPPs) at the mission level and the elements from the sub-domains that are relevant to those KPPs. 'All requirements are tradeable,' but looking at how much they contribute to the KPPs, is a different way of thinking.
    - The formalization of the Assessment Flow Diagram shows how the KPPs can be enumerated and represented in SysML, and through the MBSEPak transformed into MDAO workflows in ModelCenter (UC06)
    - We conducted two deep dive special working sessions on:
        - "Current Landscape of Ontology and Semantic Web Technology in the Design of Engineered Systems" and the use of SWT tools, such as Protégé.
        - Hands on use related to topics such as: Resource Description Framework (RDF), Terse RDF Triple Language(Turtle) syntax, Protégé (an ontology editor) for ontologies, RDF4J workbench (if network conditions allow), and concepts from Basic Formal Ontology.
11. Assess AVCE iMBE. We were asked to provide a more detailed analysis of the AVCE iMBE requirements. We initially looked at the requirements, but in attempt to do the analysis started to identify additional use cases not reflected in the model as shown in Figure 11. ARDEC then did deliver the AVCE iMBE model, and we developed a set of View and Viewpoints for the model to allow for us of MDK/DocGen. ARDEC finished the Systems Requirement Review (SRR) for AVCE iMBE, but Rick Dove joined the RT-168 research team.
    - Rick Dove has done some research through the INCOSE's Agile Systems Engineering Life Cycle Model (ASELCM) project, and specifically in terms characterized by the ASELCM Pattern of Three Concurrent Systems
    - Rick provided an analysis of both the documentation and models we received on AVCE iMBE on September 20, 2017, and provided at follow-up at the October 11th working session

2.2 WORKING SESSIONS AND SPONSOR-SUPPORTING EVENTS

A component of the research and required deliverables are conducting working sessions that inform the ARDEC 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 ARDEC; this approach has been especially important for working other SERC research tasks, such as with NAVAIR given the recent changes under SE transformation, and has been well received by ARDEC during Phase I. In addition, NAVAIR and ARDEC representatives have attended the other teams' working sessions and some of the bi-weekly meetings. For continuity and ease of access, we have included the Phase I working session information in Section 2.2.2.

2.2.1 PHASE II WORKING SESSION

A major finding from working with NAVAIR is that face-to-face working session are very effective a provide broader information to more stakeholders. We have participated 42 working session with NAVAIR dating back to 2013. We have adopted this practice with ARDEC and the following provides a summary of the Phase II working sessions:

- Working session #6: 11-October-2017 held at Stevens
  - The session covered the following topics:
    - ARDEC Update and Focus for Year 2
    - SERC RT-168 Updates Overview
    - Reflections on Working Session for Semantic Technologies for Systems Engineering (ST4SE)
    - Applications of Ontologies and Support for New Research
    - Additional Perspectives on AVCE iMBE Assessment
    - OpenMBEE and Docker, and IoIF
    - MDAO for Graphical CONOPS and ModelCenter
    - Modeling Monterey Phoenix using SysML and Semantic Web Technology
    - Decision Framework and Formalizing Assessment Flow Diagram through MDAO
- Working session #7: 14-December-2017 held at Picatinny
  - The session covered the following topics:
    - AVCE-iMBE – Current status and future work
    - SERC RT-168 Updates Overview and Surrogate Pilot
    - Overview and demonstration of OpenMBEE Model Development Kit (MDK)/DocGen versus MMS and View Editor
    - Brief Summary on Ontology Bootcamp held December 5, 2017
    - Approach to Facilitate Ontology Integration into AAMODAT
    - Plan for Ontology and SWT Breakout Session for Working Session #8
    - Update Decision Framework and Formalizing Assessment Flow Diagram through MDAO and Integrating with AAMODAT
- Working session #8: 21-February-2018 held at Picatinny
  - The session covered the following topics:
    - AVCE-iMBE – Current status and future work
    - SERC RT-168 Overview, Task Update, and Future Plans
    - Ontology Integration into AAMODAT Update and Demo
    - Update Decision Framework and Formalizing Assessment Flow Diagram through MDAO and Integrating with AAMODAT
    - Update from Research MDAO
    - Demonstration Multiple OpenMBEE Model Management System (MMS) scenarios; we are creating videos that will be posted on the SERC YouTube channel
- Working session #9: 3-April-2018 held virtually due to weather-related issues
  - The session covered the following topics:
    - AVCE-iMBE – Current status and future work
    - SERC RT-168 Overview, Task Update, and Future Plans
    - IoIF Design for AAMODAT Processing

        - MMS, ViewEditor, and SysML
        - Data Request Proxy and Data Visualization
        - SWT and Decision Layer
- Working session #10:
    - The session covered the following topics:
        - AVCE-iMBE – Current status and future work
        - SERC RT-168 Overview, Task Update, and Future Plans
        - IoIF Platform Presentation and Demonstration
        - Future IoIF Research & Application at ARDEC
        - Continuing Ontology Work to Support IoIF
        - Surrogate Pilot and OpenMBEE
- Working session #11
    - The session covered the following topics:
        - Ontology efforts for ARDEC domains
        - Pilot Project Integration of Engineering and Physics Models
        - SERC RT-168 Overview, Use Case Updates, and Future Plans
        - IoIF SWT Update & Demonstration
- Special working session on Ontology & Semantic Web Technology (SWT) Deep Dive: 2-November-2017 held at Picatinny
    - Conducted by Mary Bone, Tom Hagedorn, and Roger Blake
    - Ontology Fundamentals & Engineering Practices
        - Ontology Fundamentals
        - Current Application of Ontologies in Engineering
        - Ontology Engineering Practices & Basic Formal Ontology
    - SWT Implementation & Benefits
        - SWT Architecture and Supporting Tools
        - SWT Applications & Benefits
    - SERC Implementation of SWT
        - AAMODAT & Decision Making with SWT
        - ARDEC IoIF SWT Implementation
    - Art of the Possible and Lessons Learned
        - Current Art of the Possible
        - Lessons Learned from other SWT efforts
    - Working Group Open Collaboration Forum
        - Next steps IoIF Current SW State
        - Collaboration between ARDEC and SERC moving forward
        - Use Case development
        - Future Demos
- Special working session on Ontology & Semantic Web Technology (SWT) – Guided Tutorial of Protégé, Feb 5, 2018
    - Conducted by Paul Grogan with support from Tom Hagedorn
    - Overview of ontology concepts and tools
        - Review concept of linked data (subject -> predicate -> object)
        - Review concept of ontologies (constraints on relationships)

        - Define object classes and properties for example case in Protégé
        - Define object instances and perform querying/inferencing operations
    - Constructing ontologies leveraging existing work
        - Review concept of upper-level and other dependent ontologies (e.g. BFO, IAO, etc.)
        - Import dependent ontologies into Protégé
        - Define object classes and properties for example case in Protégé
        - Add object instances and perform querying/inferencing operations
    - Map concepts to ARDEC-specific needs
        - Whiteboard discussion of key ARDEC concepts
        - Discussion to identify key class and property constructs relevant to work

2.2.2 PHASE I WORKING SESSIONS

This section summarizes the working sessions from Phase I of this research task to make it easier to understand the combined efforts from Phase I to the transitioned focus for Phase II.

- Working session #1: 21, 22-Sep-2016 held at ARDEC
    - The SERC team provided an overview elaborated from the proposal discussing an approach to use case study scenarios to address the lifecycle concerns from CONOPS, mission and system analysis, using MDAO for tradespace analysis, Model-Based System Engineering linking to risk and the decision framework. This was presented in the context of their Digital Thread concept. The SERC team also discussed the potential synergies with NAVAIR Systems Engineering Transformation and the Digital Engineering Strategy initiative coordinated by Deputy Assistant Secretary of Defense (DASD). Discussed the concept for developing the ontology underlying the requirement manager (top-level priority)
- Working session #2: 10-Jan-2017 held at ARDEC
    - This session covered the broad objectives identified by ARDEC, to:
        - Discuss progress in research areas
        - Share lessons learned from their own efforts on Challenge Areas
        - Identify areas for enhanced collaboration
        - Engage in general model-based engineering discussions
    - A number of presentations and demonstrations from ARDEC, SERC, and NAVAIR were given to inform the audience and to stimulate further discussions, including:
        - Status of AVCE-iMBE Project – ARDEC, Cliff Marini
        - Dynamic Model Challenge Overview – ARDEC, Rich Swanson
        - NAVAIR SE Transformation Overview – NAVAIR, Jaime Guerrero
        - Overall Status of RT-168 Transforming Systems Engineering through Model-Centric Engineering - SERC, Mark Blackburn
        - Demonstration: Graphical CONOPS – SERC, Roger Jones
        - Demonstration: VT-MAK Mission Simulation – SERC, Roger Blake
        - Integrated Mission Modeling: Approach and Initial Results – SERC, Paul Grogan
        - Demonstration: Multidisciplinary, Design, Analysis and Optimization – SERC, Steven Hoffenson

        - Overview of Integrated Model Based Engineering Environment (iMBE-E) Data Challenges - ARDEC, John Campbell
        - Data Ontology/Information Model - SERC, Mark Blackburn
        - Decision Framework Approach and AAMODAT, ARDEC, Matt Cilli
- Working session #3: 30-Mar-2017 held at Stevens
    - ARDEC AVCE-iMBE Update, Cliff Marini
    - NAVAIR Progress update, Mark Blackburn
    - RT 168 Progress update, Mark Blackburn
    - Semantic Web Technologies Demo & Discussion, Mary Bone
    - Semantic Web Technologies Demo and Discussion… continued
    - USC ICT Research Presentation, Edgar Evangelista
    - MBE Tools: Syndeia, OpenMBEE, Jeff McDonald, Mark Blackburn
    - Mission-level simulation using High Level Architecture (HLA) Demo, Roger Blake, Paul Grogan
- Working session #4: 13-Jun-2017 held at ARDEC
    - ARDEC updates, Christina Jauregui, Cliff Marini, Greg Nieradka
    - OpenMBEE, Mark Blackburn
    - OpenMBEE MDK/DocGen for the AVCE model, Benjamin Kruse
    - SysML/MDAO/MBSE Analyzer, John Dzielski
    - MDAO updates, Brian Chell
    - Graphical CONOPS update and demonstration, Roger J.
    - Semantic Technology for SE Working Group/ NASA/JPL Integrated Model Centric Engineering (IMCE) Ontologies and SWT, Mark Backburn, Mary Bone
    - Integration and Interoperability Framework (IoIF) – Demonstration, Roger B, Roger J, Paul)
    - NAVAIR RT-170/RT-176 updates, Modeling for the Surrogate Pilot, Mark Blackburn
    - Requirement V&V through Monterey Phoenix (Mark Blackburn)
- Special Session: 31-July-2017 held at Stevens
    - This special session invited our sponsors from ARDEC, NAVAR, and DASD(SE), but also other organization Naval Surface Warfare Center, Digital Warfare Office, and MITRE, and industry guests from Raytheon working on Semantic Web Technologies and Ontologies
    - Objectives included: “Provide Big Picture – Mental Model”
        - Use historical context of research investigating “the most advanced and holistic approaches and technologies supporting state-of-the-art in Model Centric Engineering” aka Digital Engineering
        - Summarize expanse of research thrusts dating back to initial NAVAIR air research in 2013
        - Discuss alignment with sponsors’ evolving needs, transformation, and goals of digital engineering initiative
        - Provide awareness of collaborations with other initiatives, industry, government, academia & open communities
    - “Past – Why” – Historical perspectives – How we got here and why
    - “Present – What” - Aligning the research gaps and challenges for a Systems Engineering Transformation

  - "Future – How" - Blending and evolving our research results with Digital Engineering (DE) Transformations across the DoD to be in a Future State by Computationally Enabled DE
  - Deep Dive a Few Research Topics
  - Integrated Systems Engineering Decision Management (ISEDM) Process Enabled by Digital Engineering Technologies, presented by Matthew Cilli
  - Semantic Technologies and Ontologies Research to enable Trade Space Analytics for Engineered Resilient Systems, presented by George Ball, Raytheon
  - Breakout Session discussing
    - Risk for Digital Engineering Transformation
    - Priorities for Digital Engineering Transformation
  - Forward Planning and Actions
- Working session #5: 1-August-2017 held at Stevens
  - Perspectives on July 31 Session: Systems Engineering Transformation through Model Centric Engineering
  - ARDEC challenge updates
  - Presentation and demonstrations on IoIF overview and demonstration (UC09, UC00, UC01, UC02, UC04), and IoIF model and workflow representation
  - Overview of OpenMBEE plan for integration into the IoIF
  - Decision Framework (UC06) and Formalizing Assessment Flow Diagram through MDAO (UC03)
  - Status updates of the Graphical CONOPS (UC01) integration with MDAO (UC03)
  - Status update from UCE/ICE
  - Next steps for Phase II

A comprehensive list of the meetings, demonstrations, deliverables and ARDEC-relevant events is provided in Appendix 19J.

PART II: USE CASE DETAIL SUMMARY

The material in Part II provides additional detail on the latest status on the tasks in the context of the research use cases, including information shared during some of the working sessions and bi-weekly meetings. For additional historical perspectives there is material covered in Part II of the RT-141 final report [26], RT-157 final report [27] and RT-170 final report [29], which still provides relevant information to this research, and some of most synergistic research that is highly relevant to RT-168 has been blended into this report. At the request of our sponsor, additional details about the use cases has been created by the researchers, which is provided in Part III.

Each of these sections has a team of researchers, which are reflected by Figure 1. We are adding the information from the different perspectives and will continue to integrate the story as the research results evolves through Phase II (August 2017 – August 2018).

3 INFORMATION MODEL (UC00)

MCE is enabled by computational technologies that now provide a means for using modeling and simulation in a transformed approach to systems engineering. A key problem is that most of these technologies are not integrated (and many may never be). We know that there will be tools that provide some degree of integration, and we recommend that our sponsors opportunistically take advance of these technologies. However, our research is interested in an approach to using data interoperability as a means (or surrogate) for accomplishing integration, when tool-to-tool integration is unlikely or challenging. There have been demonstrations in working, and deep dive sessions on semantic technologies that are enablers for ontology interoperability. The most advanced research is defined in more detail under UC10 (see Section 13).

This information model characterizes the underlying information and relationships to "everything" that might need to be produced by the tools of AVCE. We are using OWL and SWT to represent the information. Our efforts with ARDEC are also complemented by our efforts with NAVAIR and the Semantic Technologies for Systems Engineering initiative (ST4SE) that was established in April 2017 (see more details in Section 15.5).

3.1 SEMANTIC TECHNOLOGIES FOR SYSTEMS ENGINEERING

Briefly, the SWTs are based on a standard suite of languages, models, and tools that are suited to knowledge representation. Figure 7 provides a perspective on the SWT stack, which includes eXtended Markup Language (XML) [146], Resource Description Framework (RDF) [200] and Schema (RDFS), Web Ontology Language (OWL) [199] (i.e., OWL2), the SPARQL Protocol And RDF Query Language (SPARQL) [201], and others. RDF can describe instances of ontologies – that is, the data for particular model instances, where OWL relates more to metamodels describing the class of information that can be characterized as RDF instances. RDFS extends RDF and provides primitives such as Class, subClassOf, and subPropertyOf. The SWT was created to extend the current Internet allowing combinations of metadata, structure, and various technologies enabling machines to derive meaning from information, both assisting and reducing human intervention. This technology is generally applicable to many different applications, and our research is beginning to reflect that from the demonstrations of the IoIF, to the Decision Framework (UC06), and communicating the uses of SWT by NASA/JPL, and how such capabilities can be integrated within a model based engineering environment, like OpenMBEE to provide additional reasoning on the information that is captured such as completeness, consistency and well-formedness.

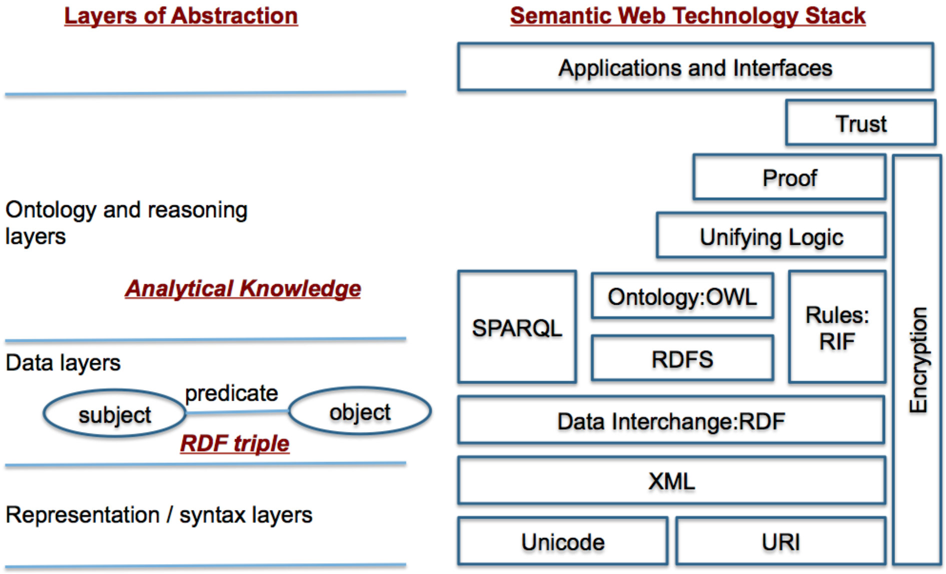
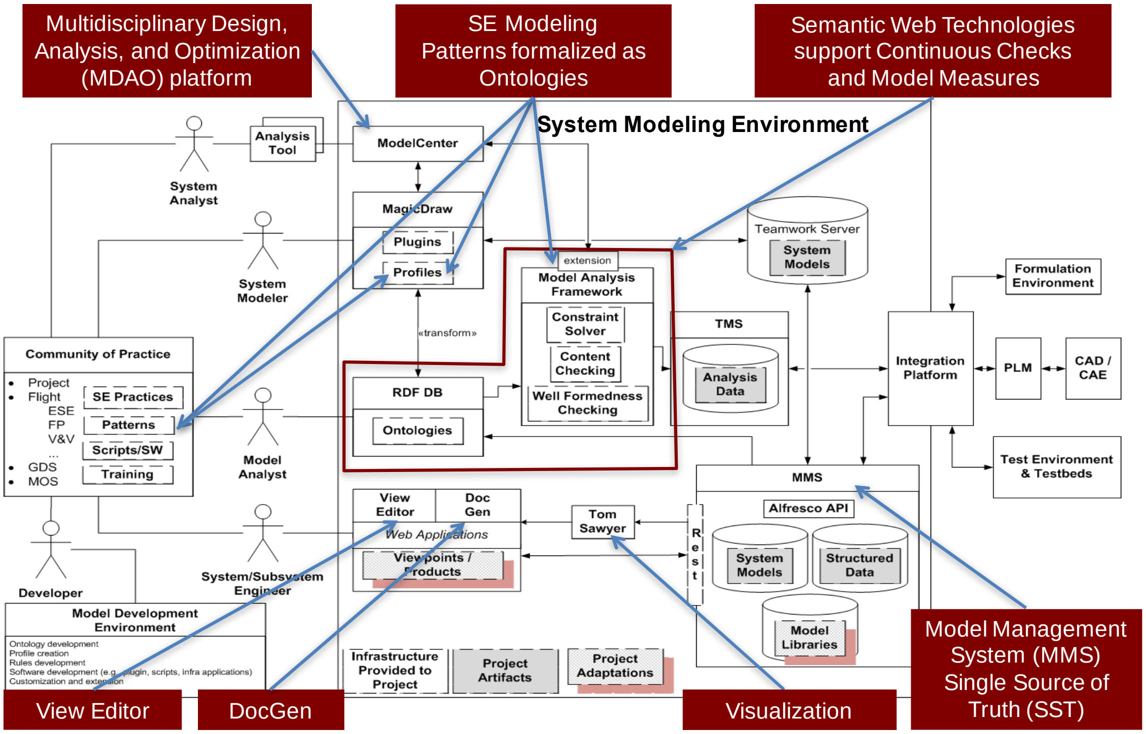


Figure 7. Semantic Web Technologies related to Layers of Abstraction

Figure 8 provides another perspective using an instantiation created by NASA/JPL, which reflects a number of the pieces we are interested in using:

- Three core elements of View Editor, DocGen and Model Management System (MMS)
- MagicDraw client (in which the MDK/DocGen) plugin works
- Teamwork Cloud server from NoMagic is used with MMS
- The NASA ontologies for Systems Engineering used to check constraints (e.g., consistency, completeness, well-formedness) [102] related to the model is shown in Figure 9
  - These are open-sourced
  - We are opportunistically leveraging these capabilities both with ARDEC and NAVAIR through our efforts with the ST4SE
  - These ontologies have grown out of a history of work, including the INCOSE modeling patterns group



*An Integrated Model Centric Engineering (IMCE) Reference Architecture for a Model Based Engineering Environment (MBEE), NASA/JPL, Sept, 2014.

Figure 8. NASA/JPL Instantiation of OpenMBEE (circa 2014)

The following figures have been taking from Model-Centric Engineering, Part 3: Foundational Concepts for Building System Models [103]. Figure 10 shows the Integrated Model Centric Engineering (IMCE) concept that is being developed. The process involves:

- Creating ontologies for foundational systems engineering derived from the modeling patterns (reflected in Figure 9)
    - This can be done in any OWL modeling tool such as the open source Protégé
    - The ontologies are turned into SysML profiles
    - The SysML profiles are loaded into a modeling tool for creating models
    - The profiled SysML models are exported back into OWL statements
    - Checks for completeness, consistency and well-formedness can performed

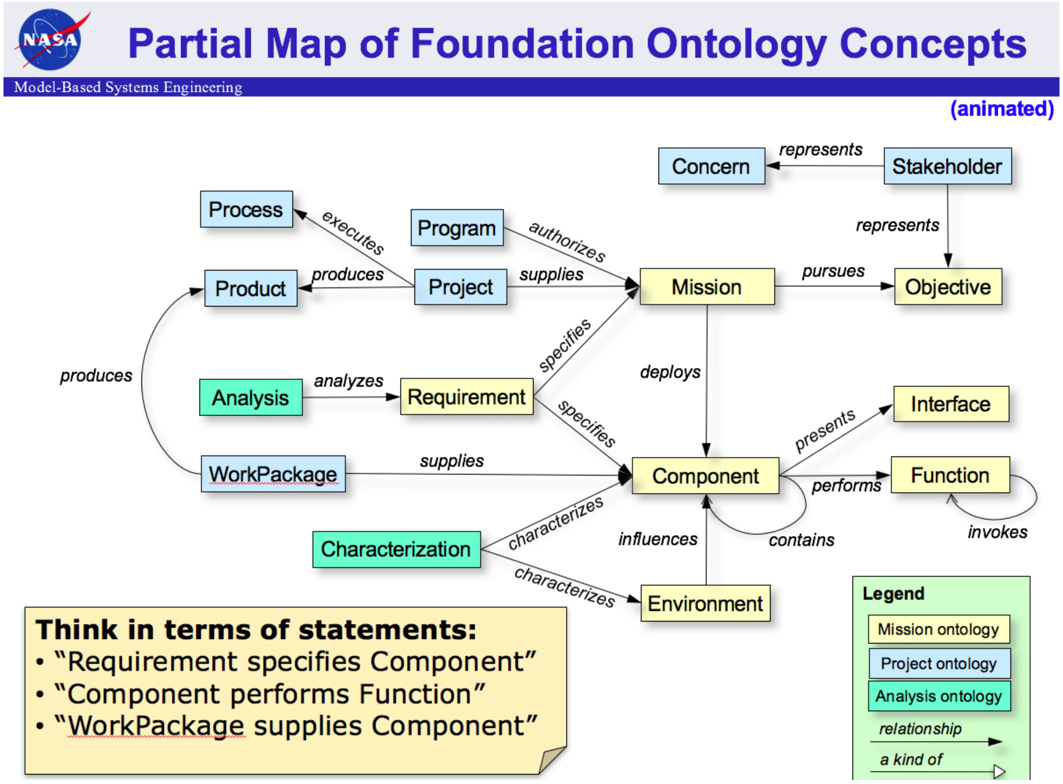


Figure 9. NASA/JPL Foundational Ontology for Systems Engineering

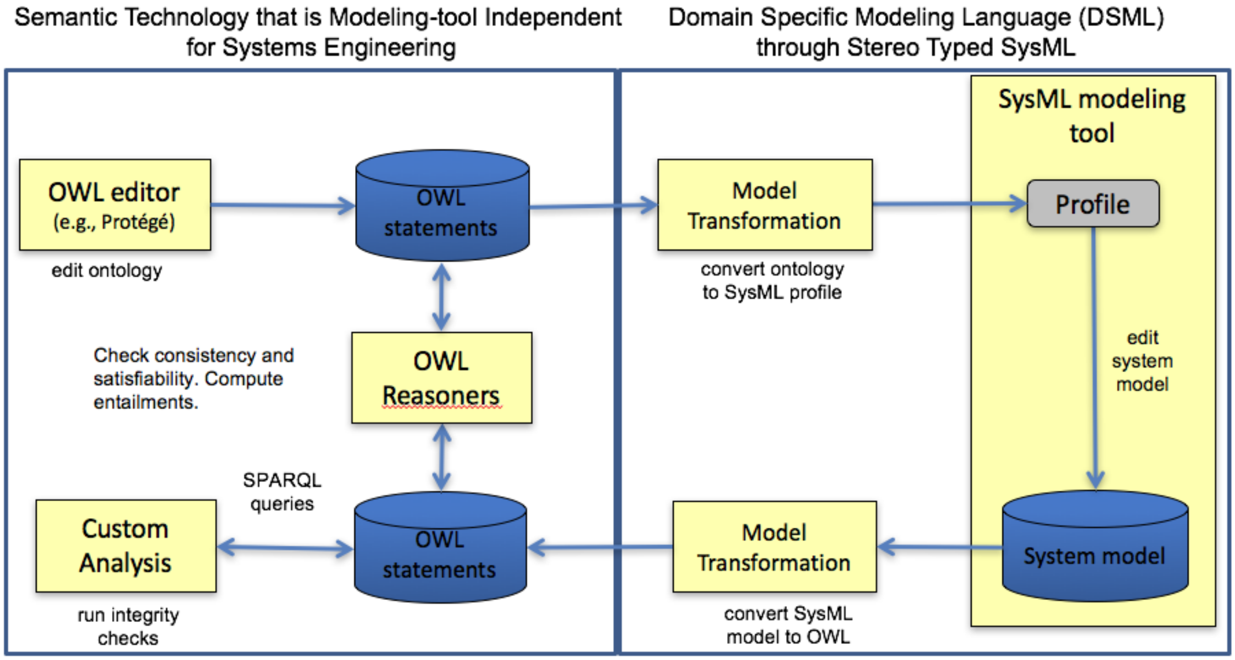
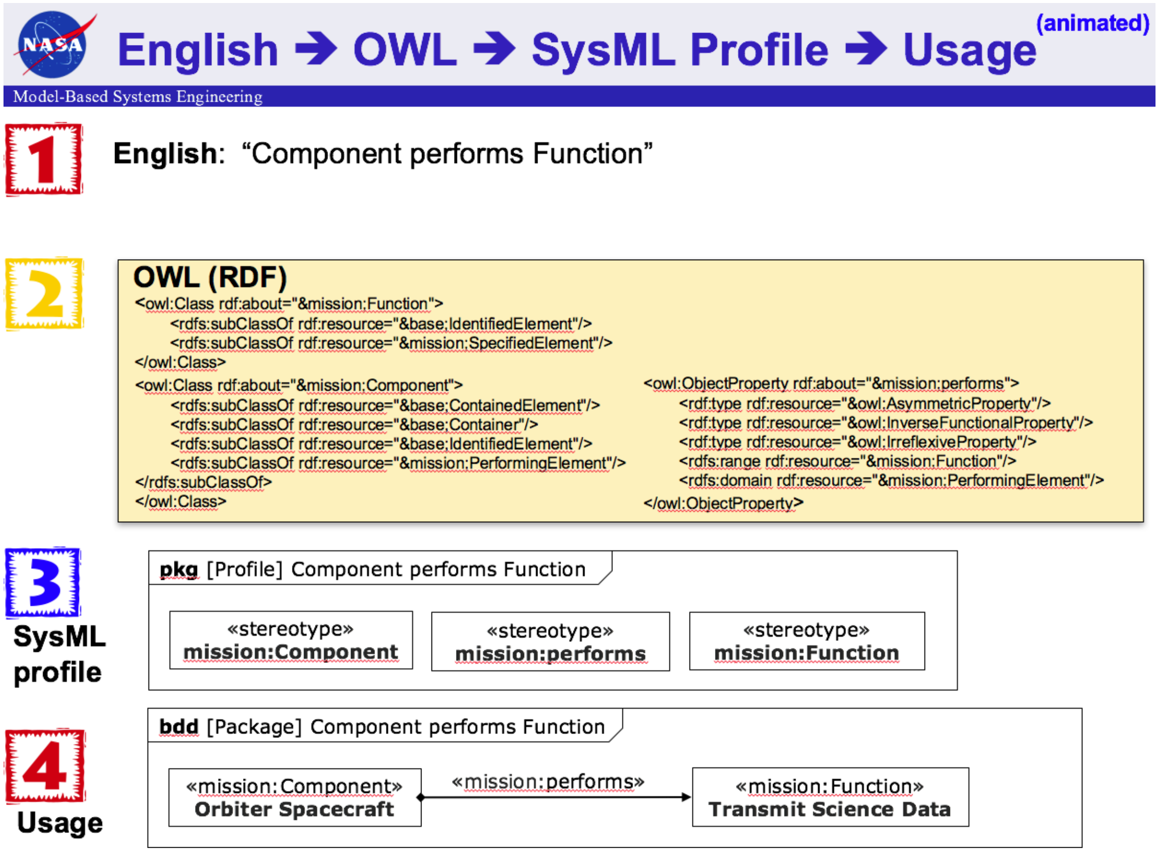


Figure 10. From Ontologies to SysML Profiles and Back to Analyzable OWL / RDF

Figure 11 shows the various representations associated with the concept described in Figure 10:

1. The modeled statement in English is: "Component performs Function"
2. The OWL/RDF representation of the statement in low-level XML for this same statement
3. The Profile and Stereotypes used in the model (loaded into a SysML model)

4. The Stereotypes used in a SysML Block Definition Diagram (BDD)



English → OWL → SysML Profile → Usage (animated)

Model-Based Systems Engineering

1 **English**: "Component performs Function"

2 **OWL (RDF)**

```
<owl:Class rdf:about="&mission;Function">
      <rdfs:subClassOf rdf:resource="&base;IdentifiedElement"/>
      <rdfs:subClassOf rdf:resource="&mission;SpecifiedElement"/>
</owl:Class>
<owl:Class rdf:about="&mission;Component">
      <rdfs:subClassOf rdf:resource="&base;ContainedElement"/>
      <rdfs:subClassOf rdf:resource="&base;Container"/>
      <rdfs:subClassOf rdf:resource="&base;IdentifiedElement"/>
      <rdfs:subClassOf rdf:resource="&mission;PerformingElement"/>
</rdfs:subClassOf>
</owl:Class>
```

```
<owl:ObjectProperty rdf:about="&mission;performs">
      <rdf:type rdf:resource="&owl;AsymmetricProperty"/>
      <rdf:type rdf:resource="&owl;InverseFunctionalProperty"/>
      <rdf:type rdf:resource="&owl;IrreflexiveProperty"/>
      <rdfs:range rdf:resource="&mission;Function"/>
      <rdfs:domain rdf:resource="&mission;PerformingElement"/>
</owl:ObjectProperty>
```

3 **SysML profile**

pkg [Profile] Component performs Function

«stereotype» **mission:Component**

«stereotype» **mission:performs**

«stereotype» **mission:Function**

4 **Usage**

bdd [Package] Component performs Function

«mission:Component» **Orbiter Spacecraft**

«mission:performs»

«mission:Function» **Transmit Science Data**

Figure 11. Multiple Representations in Process

3.2 CHALLENGE OF CROSS-DOMAIN MODEL INTEGRATION

We believe that SWT has the potential to contribute significantly to many of the user cases. As reflected in Figure 3, the challenge is to characterize this information for each of the various domains, including requirements, risks, designs (e.g., electrical, mechanical, etc.), and analyses. The challenge of cross-domain modeling integration can be illustrated using the following example. While an aircraft may have thousands of objects, consider the relationships for a refueling value of a UAV, as shown in Figure 12. There is one object discussed in this example (i.e., Valve), however, there are many domains that bring in cross-domain relationships to that Value, along with other objects, such as:

- Mechanical **Domain**
    - Valve connects to a Pipe
- Electrical **Domain**
    - Switch opens/closes Value
    - Maybe using a combination of hardware and software
- Operator **Domain**
    - Pilot remotely sends message to control Value
- Communication **Domain**

    - Messages sent through networks: 1) within the aircraft system, and 2) from the remote operator
- Fire control **Domain**
    - Independent detection to shut off Valve
- Safety **Domain**
    - Looks top-down at potential hazards through Fault Tree Analysis (FTA)
    - Looks bottom-up using Failure Models and Effects Analysis (FMEA) to analyze failure impacts from specific designs of components

Figure 12. Example of Cross Domain Relationships Needed for System Trades, Analysis and Design

A problem is understanding the cross-domain impacts of designs and analyses that might be needed if one object within these related domains change. In general, there are different tools used in different domains, and the tools are often not integrated, nor are they able to share semantically-relevant data. Tool integrations are often dynamic consequences of customer requirements to continue improving the tools, thus the tools are constantly being updated, which further adds to the challenge of tool-to-tool integration.

The use case diagram in Figure 1 reflects why there are so many associations from the other use cases. In addition, we have shown that it is technically feasible to capture this information and provide it as input to the Decision Framework as demonstrated in the later version of IoIF as shown in Figure 6. The research approach provides an initial demonstration of the use of SWT to both characterize the data and information as well as rules, and query language for processing and data exchange. Several briefings on SWT concepts (e.g., ontologies) and example uses have been provided in several working sessions, deep dive sessions and webinar sessions.

We are evolving an IoIF as part of UC09 as shown in Figure 6. We are working with other use case teams to provide a demonstration of Decision Framework enabled by semantic technology (UC00). We have a decision ontology, in OWL, which we believe provides support using SWT and Decision Layer (UC10). In collaboration with NAVAIR and NASA/JPL, we would also like to bring in the IMCE ontologies for systems engineering. We are considering using tool-to-tool integration as discussed in UC09, Data Acquisition and Aggregation in research to integrate Graphical CONOPS (UC01), and Mission and System Operational Capabilities (UC02).

3.3 UC00 MAPPING TO OTHER USE CASES

We plan to continue research in the other two areas of Graphical CONOPS (UC01) and Mission and Systems (UC02) to understand the flow of information needed to be linked between them, and characterize those linkages in an Information Model. The information produced under the following use cases has begun to characterize elements of the metamodels, for example:

- Parameters in the Graphical CONOPS
- High Level Architecture (HLA) metamodel for both VT MAK [115] and Distributed Simulation

Use cases UC03, UC04, and UC05 involve the need 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. At the system level they may be developed using MBSE methods and be represented in standard modeling languages such as SysML [148]. The linkages between the MBSE and design disciplines, usually referred to as Model-Based Engineering (MBE), 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 [44]. We believe there are opportunities to address this need in more tool agnostic ways using SWT. See UC09 and UC10.

A key reason for the need for cross-domain model integration is the underlying complexity needed to accomplish the scenarios associated with Figure 13. In addition, our research as illustrated by the DARPA META project [9] 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 methods, 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.

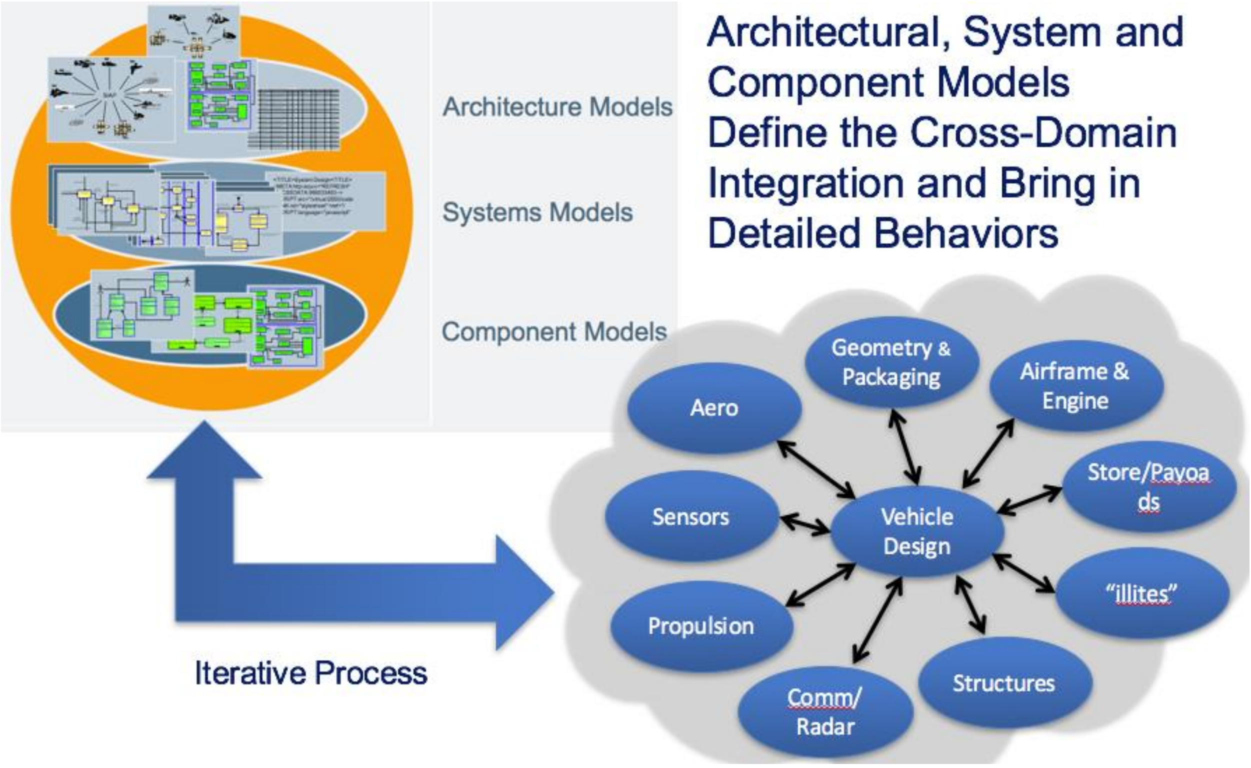


Figure 13. Integrate Multiple Levels of System Models with Discipline-Specific Designs

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 the 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 14 [58], there can be a very large set of tools that can be used to develop the needed data and information across all of the domains. Notionally the Reference Technology Platform (RTP) [5] is the collective set of tools that an organization has in their inventory. Any specific program creates a RTP instance. A key challenge is integrating the assembled tools, especially when they may not have been created to be integrated, and equally important is that the methods for assembling and using these analysis workflows is largely in the heads of a few subject matter experts, as explained by our sponsors. 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 a leading-edge approach by NASA/JPL is to ensure models are created that comply with established modeling patterns that have been formalized using ontologies. We provided information on the NASA/JPL approach, which transforms the model information into a tool-neutral AST based on ontologies, and then uses standard SWT to apply checks to ensure completeness and consistency [102].

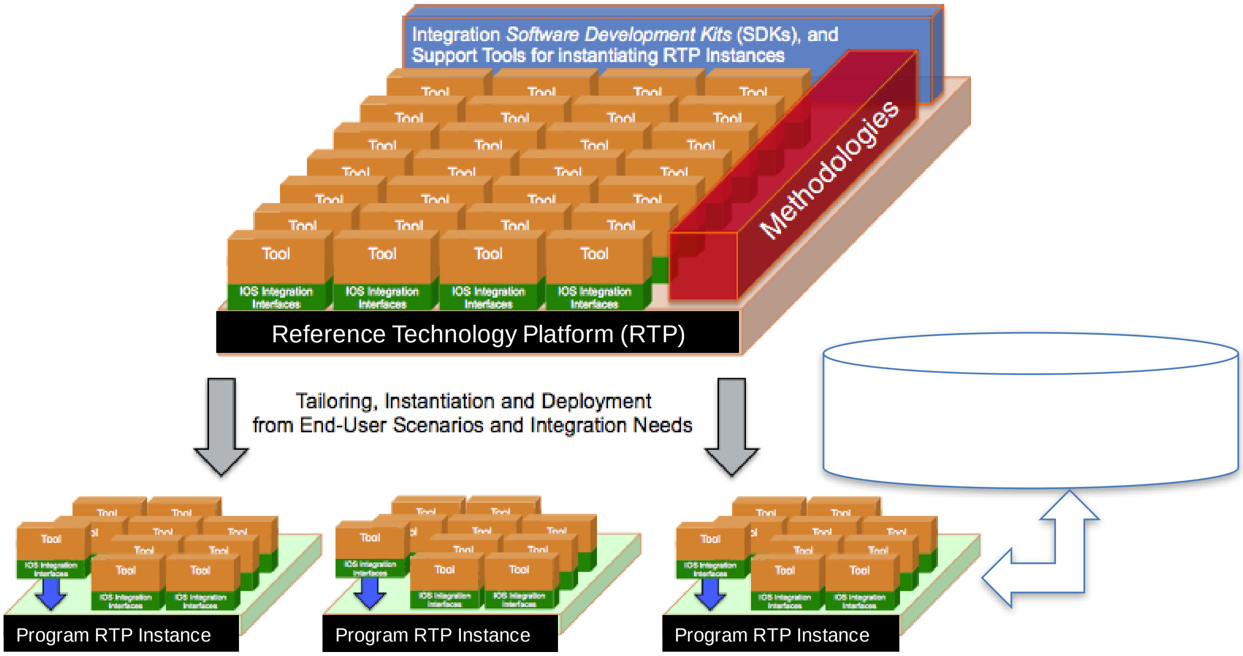


Figure 14. Appropriate Methods Needed Across Domains

3.4 UC00, UC07 AND UC10 (DECISION FRAMEWORK IN IOIF)

As shown in Figure 1, this task relates to UC07 and UC10. We have demonstrated (see UC06) that the assumption by Dr. Matt Cilli that the needed inputs to the Decision Support Model Construct [49] (referred to as Decision Framework) can be captured in models. This concept and process has been demonstrated to provide senior management and program managers with visual representation of key tradeoff defined in terms of Performance, Cost, Time and Risk.

4 GRAPHICAL CONOPS (UC01)

There are 11 different modeling and simulation examples that are being developed to support UAS and Counter UAS analysis case study. These different approaches involve different researchers, and look at the problems using different technologies, both in terms of types of abstractions, level of fidelity, no human-in-the-loop, and humans-in-the-loop, which also have an impact on trading off cost and value of the simulation. Each approach is described in the subsection below. Fundamentally, we are also interested in the information (metamodels, which map to OWL) and associated methods to produce and analyze this information in order to integrate with the other models in use cases UC01, UC02, UC03, UC04, and UC05.

The latest scenario investigates an extension of the prior work to using the Graphical CONOPS technologies Unity gaming engine with Multidisciplinary, Design, Analysis and Optimization (MDAO) (UC03) using ModelCenter. The team has extended the prior demonstration extending the initial autonomous UAS capability [23]. There have been updates since Working Session #6 that allow the simulations to run faster than real-time.

4.1 UAV CONOPS USING GAMING ENGINE SIMULATION (JONES VIEW)

Engineered systems have advanced to the stage in which they share many properties with biological and sociological systems. Engineered systems can have systems embedded in them, and those subsystems can have subsystems embedded in the subsystems. This is reminiscent of the layered level of complexity in biology. Molecular processes form cells; cells form organs; organs form organisms; and organisms form societies. In some cases, engineered systems are a part of sociological systems. A city is a combination of a social system and many engineered systems, from traffic systems to the power grid.

Nature has solved many of the problems that systems engineers are struggling with. These problems include incompatibility of systems, multidisciplinary integration, incompatible time scales, systems of systems, and more. Can we examine the manner in which Nature solves many of these problems to inform the design and optimization of complex engineered systems? This use case addresses at least this question.

Biological and sociological systems are not designed in the traditional sense. The designs emerge from interaction with each other and with the system environment through a process of evolution and natural selection.

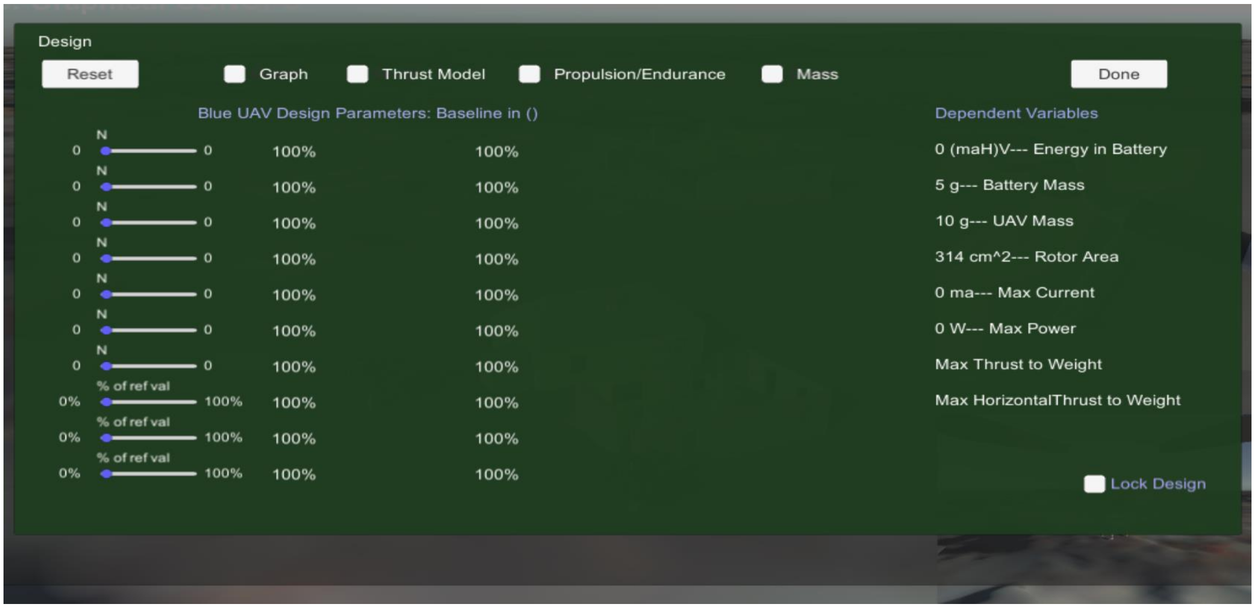
The goal of this research is to identify a general systems framework that can be used as a backend for Graphical CONOPS in support of MDAO as well as provide inputs to other types of modeling and simulation, such as both 2D and 3D approaches to mission and system simulation. Since Nature has solved many of the systems problems, the framework will be organically-based. The framework will be able to create models of a very large class of systems and systems-of-systems. As shown in Figure 15, we have created an example that has demonstrated the use of this concept in an environment involving UAS mission scenarios using the Unity Gaming Engine; this will be the canonical example.

Roger Jones has demonstrated a Graphical CONOPS created using the Unity game engine that that provides Monte Carlo simulation feedback to MDAO. There are two possible surveillance missions for a blue quadcopter. In scenario one, the blue quadcopter searches for an object, and mission is unimpeded. In the second mission, a red quadcopter actively tries to prevent the blue copter from succeeding at its mission, as shown in Figure 15. Both quadcopters are fully autonomous. There are options to change different parameters related to the two UAVs in a dynamic manner. As shown in Figure 16, there are also tabs that can be used to parametrically modify the capabilities of the two different UAVs.

- The latest version delivered to ARDEC provides features:
    - Communication with other software through JSON files
    - Uses MDAO which writes to JSON that is read by the Unity gaming engine
    - Has more realistic battery and flight models
    - Enhanced design interface that allows user to quickly explore design space around an optimum determined by static MDAO software
    - Analysis and optimization modules are integrated with ModelCenter through JSON files
    - Integrate a synchronized simulation with the output from the graphical CONOPS being published through the SWT and be consumed (subscribed) through the SWT by the 2D simulation

- ○ Demonstrate integrated simulation as part of the IoIF

Figure 15. Unity Gaming Engine Simulation of Two Moving UAV with Camera



☐ Graph ☐ Thrust Model ☐ Propulsion/Endurance ☐ Mass

Blue UAV Design Parameters: Baseline in ()

N	0	0	100%	100%
N	0	0	100%	100%
N	0	0	100%	100%
N	0	0	100%	100%
N	0	0	100%	100%
N	0	0	100%	100%
N	0	0	100%	100%
% of ref val	0%	100%	100%	100%
% of ref val	0%	100%	100%	100%
% of ref val	0%	100%	100%	100%

Dependent Variables

0 (maH)V--- Energy in Battery
5 g--- Battery Mass
10 g--- UAV Mass
314 cm^2--- Rotor Area
0 ma--- Max Current
0 W--- Max Power
Max Thrust to Weight
Max HorizontalThrust to Weight

☐ Lock Design

Figure 16. Unity Gaming Engineering Simulation MDAO

4.2 WRAPPING GRAPHICAL CONOPS WITH MULTIDISCIPLINARY DESIGN, ANALYSIS AND OPTIMIZATION

Several versions of an example that wraps the Graphical CONOPS technologies Unity gaming engine with MDAO (UC03) using ModelCenter to drive thousands of tradespace examples version 10s that would be run manually as shown in Figure 17. The capabilities covered include objective to understand and overcome the challenges for a fully automated MDAO at the Graphical CONOPS level.

The performance is measured by degree of success of a mission. It was realistic, but the simulated environment that includes counterparties was observed to behave in a surprising manner that included emergent behavior. This may have occurred, because the autonomous simulated UAS did have Artificial Intelligence (AI) that is applied to counterparties so that they can adapt to and learn behavior of system. The simulation was fully automated, that is there was no humans in the loop, except for validation of behavior. The software communicates programmatically through file transfer, as opposed to being directed manually using the parametric controls shown in Figure 16. The Monte Carlo results in thousands of runs versus 10s of run when done manually; this initial state is random and statistics for each run are captured. The simulation can run at high speed to maximize statistics and in real time to allow for human validation of simulation behavior. A second version runs the simulation faster than real-time. The simulation extended and refactored the prior Unity prototype, where:

- The blue UAV searchers for a treasure chest
- The red drone is supposed to prevent the blue drone from finding the treasure box

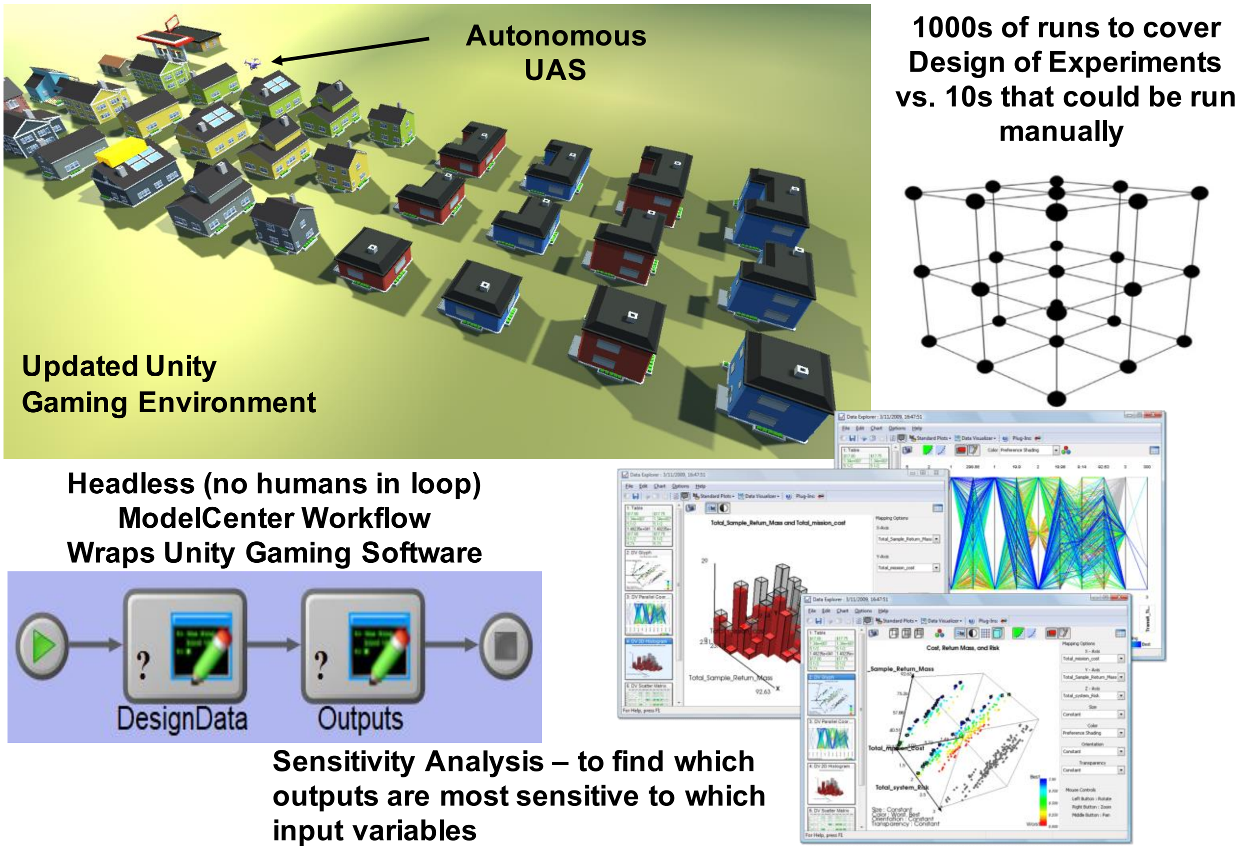


Figure 17. Explore the Integration of Graphical CONOPS Simulation with MDAO Tools

Additional findings include:

- The initial attempt was to create the simulation and strip out the visualization details to populate Phoenix Integration ModelCenter

- The architecture of the simulation was not enabled to operate in batch modes
- The software had to be re-written to work with ModelCenter
- When the simulation is running, the human cannot make edits
- The re-written simulation can be wrapped and 1000s of design of experiments (DoE) have run using ModelCenter
- It is determined that the simulations can be run faster than real-time
- MDAO can be used to optimize for system-level mission success to study far more trades than can be performed manually
- This demonstration provided some good insights
  - It is a possible opportunity for Threat Based Offset to use the mission-level MDAO model to understand emergent behavior (i.e. can you visualized the simulation and re-run it?)
  - May be possible to do this if one keeps the seed for the random-number generator
  - Other modifications are still under investigation

This capability provides a demonstration that MDAO can be applied at the mission level to graphical CONOPS, and to perform comprehensive trades using design of experiments that provides 1000s vs. 10s of runs for more systematic alternative analysis at the CONOPs level.

4.3 GRAPHICAL CONOPS (USC ICT – RICHMOND VIEW)

The USC Institute for Creative Technologies (ICT) provided support in Phase I for this use case by investigating various aspects of Early Synthetic Prototyping (ESP) capability that has been developed for RDECOM-ARDEC. They, too, use the Unity gaming platform with other technologies that integrate and study humans-in-the-loop. The scope of work includes, but is not limited to:

- Visualization of tradespace and alternatives
- Graphical CONOPS improvements
- Assess collaboration opportunity with TRADOC's ESP
- Provide recommendations for Collaborative Design Infrastructure
- Methods for logging human-game interactions

For more details on the results, please refer to the RT-168 Final Technical Report for Phase I [28].

4.4 SIMULATION TECHNOLOGIES FOR GRAPHICAL CONOPS (GROGAN VIEW)

Graphical CONOPs engages stakeholders in an interactive, immersive environment to develop a CONOPS [55] [109] [131]. It aims to improve communication between users and developers by providing a common platform on which to express issues, similar to the concept of a single text in negotiation [158].

Another element of this research is investigating the use of standard simulation technologies for graphical CONOPS. Standards are crucial to enable interoperability and data exchange across model boundaries. The two most common standards for distributed simulation are IEEE Std. 1278 Distributed Interactive Simulation (DIS) [96] and IEEE Std. 1516 High Level Architecture (HLA) [97]. DIS defines common data structures (protocol data units, PDUs) which are exchanged between simulation members in real time. HLA defines a common application programming interface (API) to a runtime infrastructure (RTI) which manages data exchange and time synchronization among simulation federates. Other related standards include IEEE Std. 1730 Distributed Simulation Engineering and Execution Process (DSEEP) [98], SISO Std. 001 Real-time Platform Reference Federation Object Model (RPR FOM) [179], SISO Std. 007 Military Scenario Definition Language (MSDL) [180], and SISO Std. 011 Coalition Battle Management Language (C-BML) [181].

In contrast to other combat modeling activities and broader military operations research (the typical application of the above standards), graphical CONOPS directly supports system design activities and, as such, does not incorporate as much detail. Instead, it seeks to identify fundamental characteristics of the target problem. The outcome of a graphical CONOPS activity produces a set of scenario parameters to describe the environment in which a system will be used. In addition, we investigate a potential interface between a SysML model and an integrated mission model.

To support this research another capability has been created and demonstrated. This is a simple scenario with UAV, Counter-UAV as a two-dimensional model of a two UASs, one “friend” and the other “foe,” with emphasis on distributed simulation using HLA to synchronize model state across simulators using internal interface within the mission model.

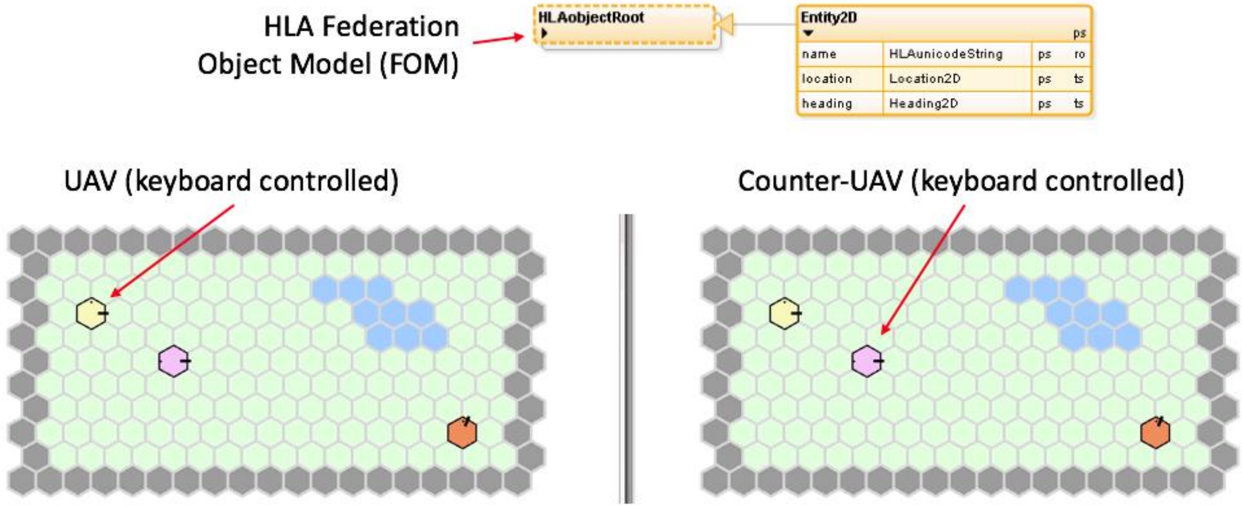


Figure 18. Mission Model using High Level Architecture (HLA) to Enable Distributed Simulation

4.5 MISSION MODELING USING HIGH FIDELITY SIMULATION VT MAK (ROGER BLAKE)

We have also secured an academic license for the VT MAK / VR-Forces tool as a high-end alternative to the two-dimensional simulation discussed in Section 4.4. VR-Forces is a high fidelity simulation environment that implements Computer Generated Forces (CGF) and a Simulator Development Environment using a HLA framework. VR-Forces contains a multitude of federate models that can be used to create interactive simulation environments to analyze various situations and behaviors of desired scenarios. We have demonstrated the use VR-Forces as a tool in our research in order to show the effects of our research and implementations. Since each VR-Forces federate model can be communicated with using a Lua [113] scripting language, we can change model parameters flexibly. The idea is that as the design tools change value, we can theoretically enter the new design parameter values into the simulation models to observe the new behaviors within the high fidelity simulation scenario. This again provides another way to use MDAO to consider different optimization (see Section 6).

We developed a demonstration for a simple UAV simulation. This is being expanded into a counter UAV mission. The scenario that we demonstrated was one which included a UAV that was scanning various entities that it encountered as shown below in

Figure 19. As we continue to build this scenario, we plan to include counter measures to the UAV like a Surface-to-Air Missile System also shown below. As the UAV flies to, and around its targets, nearby Surface-to-Air Missile Systems will fire on the UAV if the UAV flies into their kill zones as demonstrated by the green RADAR beams that illustrate the area of coverage in the Surface-to-Air Missile System shown below in Figure 20 and Figure 21.

Figure 19. UAV Scanning Targets

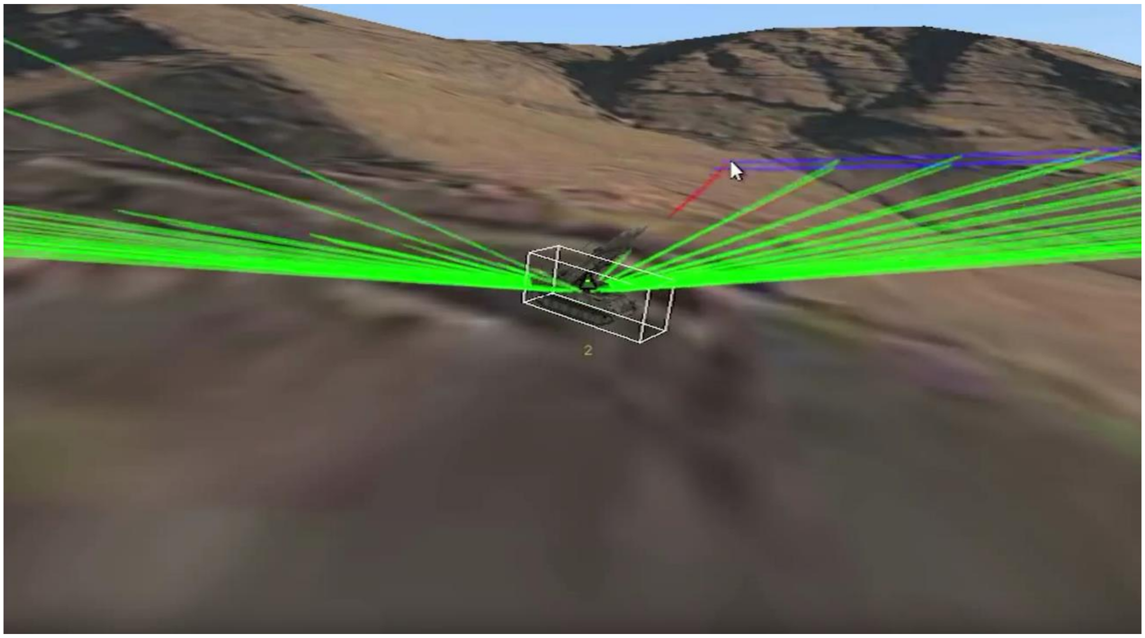


Figure 20. Surface-to-Air Missile System

By furthering this research, we hope to be able to use a publish/subscribe system that is implemented in the IoIF that utilizes tool proxies to aggregate design tool data which can be routed to the recipient design tool through the implementation of an ontology layer. By doing this, we hope to be able to facilitate the transfer of design tool parameter data though this network by using the SWT layer as the control point that decides where design parameter data is needed. We can then link the design parameter data into the federate model in our simulation to be able to observe the new model behavior in the simulation environment based on the new design tool parameter changes.

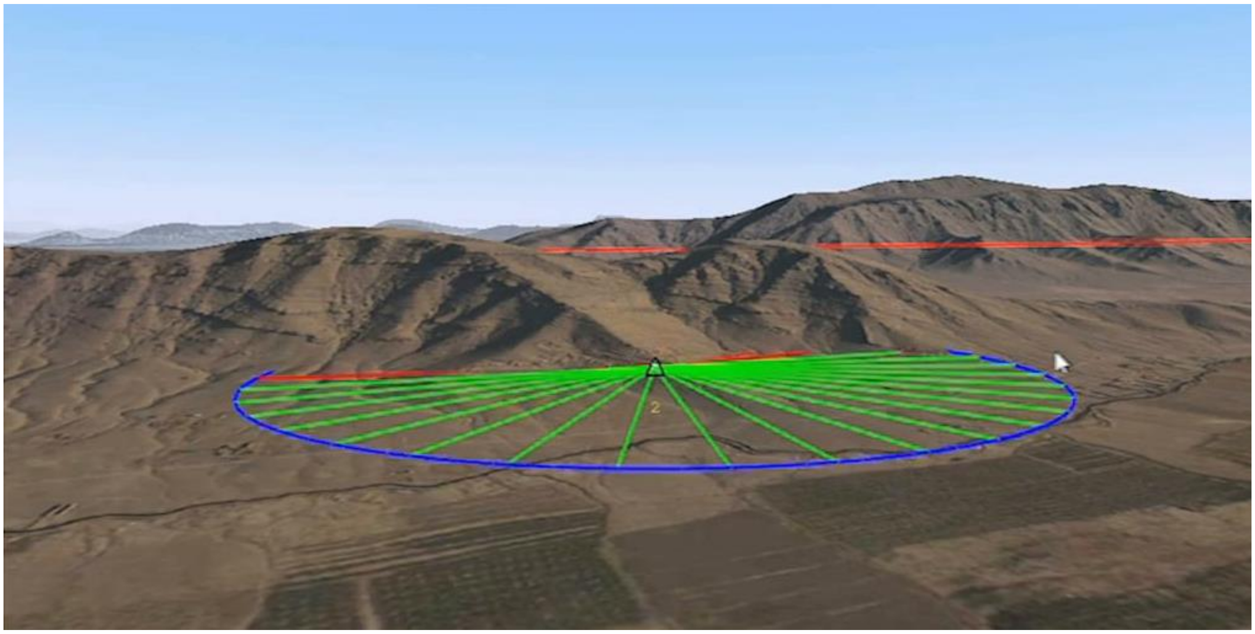


Figure 21. Surface-to-Air Missile System Area of Coverage

4.6 SURROGATE PILOT CONOPS AND MISSION MODELS

We did use the VR-Forces capabilities to develop a CONOPS for a Search and Rescue scenario for a hypothetical system called Skyzer. Skyzer has a CONOPS for an UAV that provides humanitarian maritime support use cases as reflected in Figure 22. More details can be found as part of the NAVAIR Systems Engineering Transformation on the All Partners Network (APAN) (apan.org). Several of our ARDEC sponsors are members of APAN.

Figure 22. Graphical CONOPS for Skyzer UAV

One of the synergies between the ARDEC and NAVAIR research is our ability to share information and examples. In this particular case, the graphical CONOPS for the Skyzer has been translated into an evolving mission model that is based on an Integrated Capability Framework (ICF) Operational Concept Document (Version 3.2) 22 February 2016. This document is considered “Distribution D,” which means it may only be available to companies that are doing business with the government. However, the Skyzer Mission model is available publically on the Amazon Web Services server in OpenMBEE. This approach demonstrates that modeling can be used and comply with existing standards that traditionally have been document-based.

The guidelines include:

- Thoroughly define required mission capabilities, measures of effectiveness, and associated operational conditions and constraints
- Identify System of Systems (SoS) interfaces and measures of performance through structured decomposition of required mission capabilities
- Provide a common, Cross-Systems Command (SYSCOM)/Program Executive Office (PEO) framework to facilitate enterprise level engineering across the SYSCOMs and enable efficient system integration and effective force interoperability
- Establish enterprise data structures and implementation guidance to enable iterative development of enterprise architectures
- The consistent implementation of Integrated Capability Framework (ICF) practices and guidance across assessments and stakeholders supports:
    - A common understanding of mission requirements and a structured process to identify and align systems and platform(s) capabilities to support missions

- System and platform owners with a thorough set of interoperability requirements and knowledge of what platforms, interfaces and behavior to which they need to design, along with associated standards

We have a View and Viewpoint hierarchy that extracts information from the Skyzer Mission model to "generate a specification," which aligns with the guidelines of the ICF using the OpenMBEE DocGen. A portion of the View and Viewpoint hierarchy is shown in Figure 23.

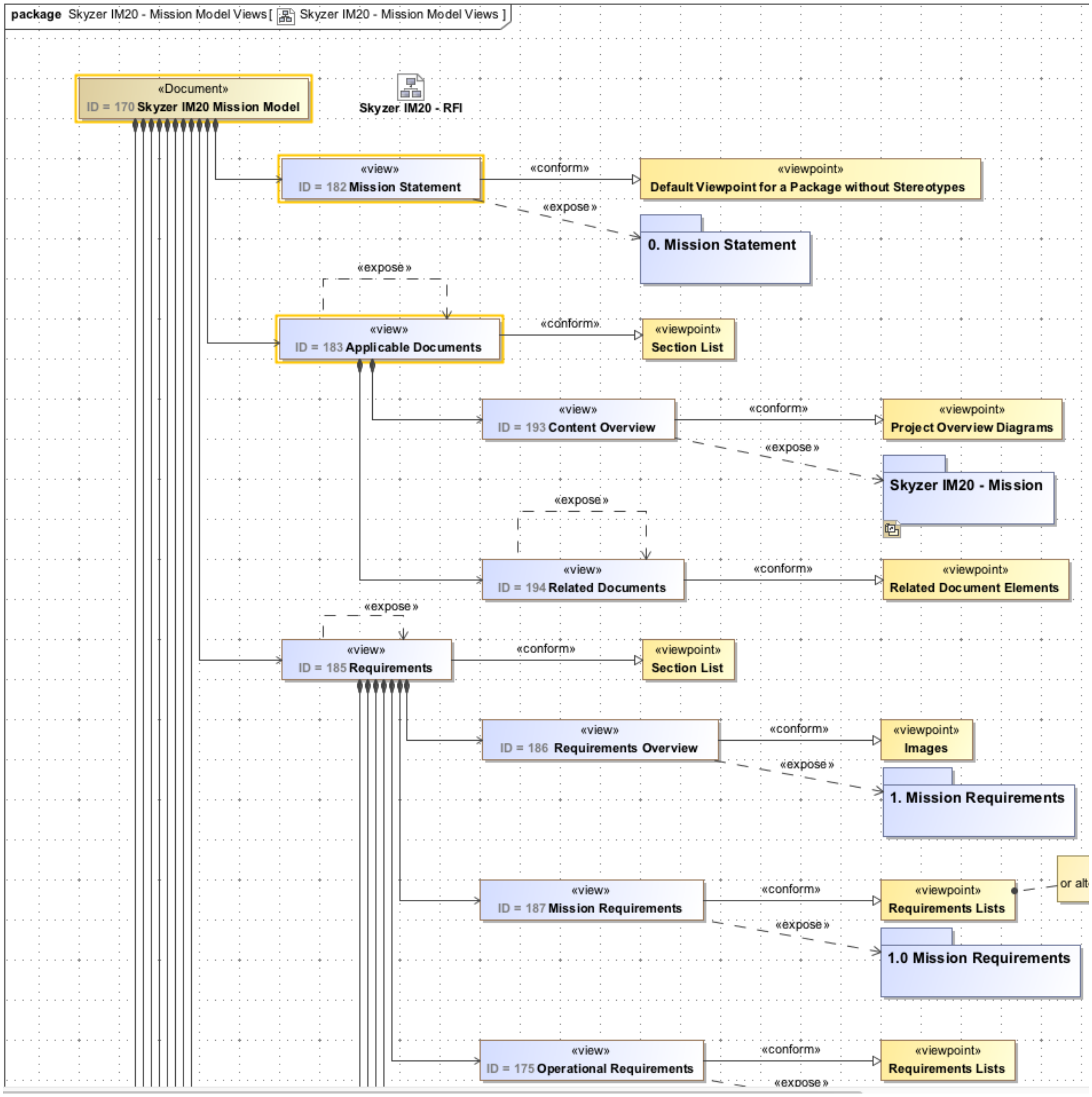


Figure 23. View and Viewpoint Hierarchy for Surrogate Pilot Mission Model

5 MISSION AND SYSTEM CAPABILITY ANALYSIS (UC02)

A mission model is a dynamic simulation model which evaluates the application of a system in the context of a scenario. It simulates the system operation to integrate and compute key performance metrics (KPMs) and assess system value over operational timescales. A mission model may either be controlled manually or executed autonomously provided adequate behavior scripting. The system model evaluates static functional capabilities for a particular system design. A system model evaluates and optimizes functional capabilities for a set of objectives and constraints.

This section extends the research discussed in Section 4 to investigate automatic transformation and exchange of data between the mission model, graphical CONOPS, and system model. As reflected in Figure 24, inputs to the mission model include scenario parameters and system functional capabilities. KPMs output by the mission model can be used to revise and alter scenario definitions and system designs as needed.

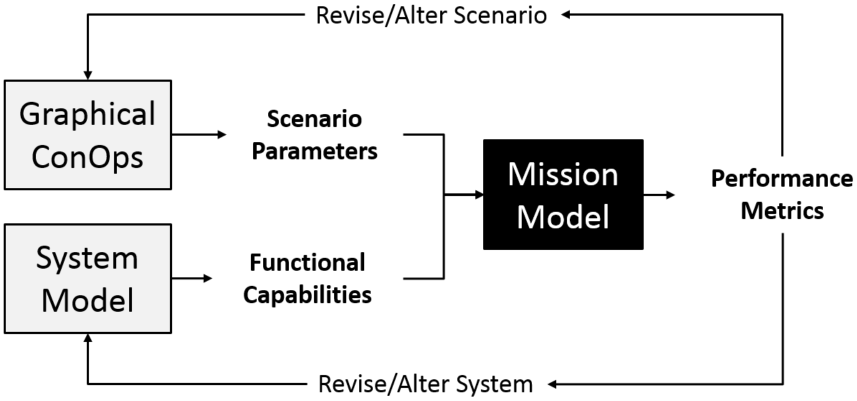


Figure 24. Scenario Parameters and Functional Capabilities are inputs to a Mission Model Which Computes Performance Metrics

This project uses an application use case scenario to study the MCE approach described above. This notional case is purposefully simplified to allow rapid modeling without proprietary or sensitive details, as discussed in Section 4.4. The use case scenario considers the conflicting operations between a UAV and a counter-UAV system. Both platforms exist in space and are equipped with sensors and engagement devices.

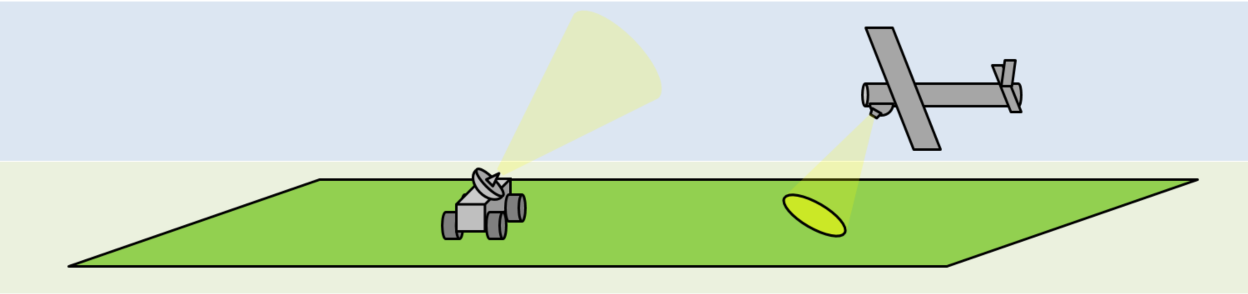
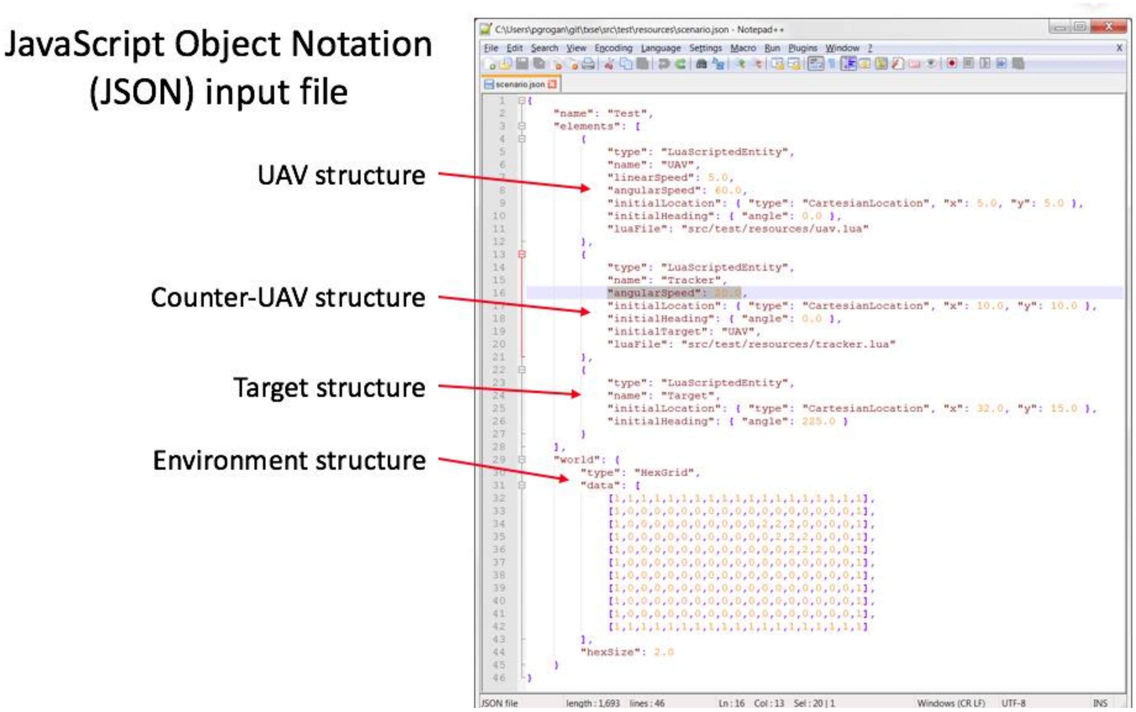


Figure 25. UAV and Counter-UAV Systems Participate in the Scenario.

Initial work has focused on development of a simplified mission model for the UAV/Counter-UAV scenario described above. The mission model is a Java executable which imports scenario and system information from external interfaces. Context parameters defining the spatial region are loaded from JSON file. System parameters defining the functional capabilities (max speed, etc.) are also loaded from JSON file and system behaviors can be expressed Lua scripts conforming to an internal API.

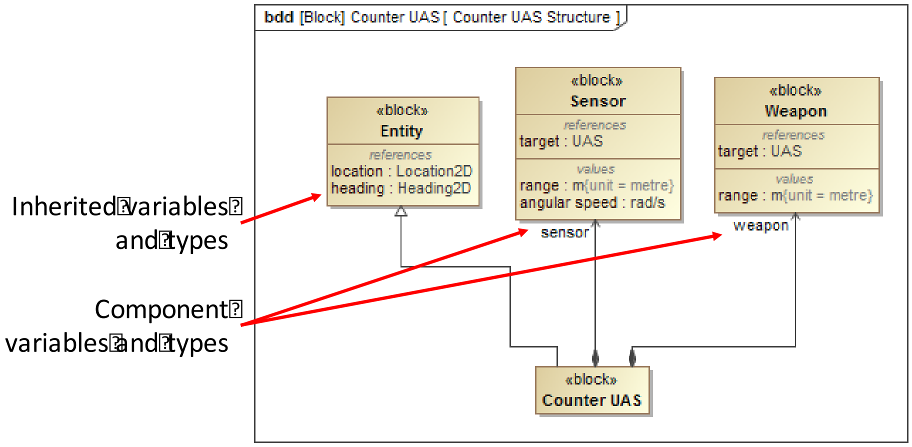
5.1 MISSION MODEL MAPPING TO SYSTEM MODEL

Paul Grogan investigated creating a representation in SysML and mapping the parameters from the simulation into SysML. We use the mission model and can extract out data about individual system elements, as well as environmental information. An example of the structural aspect of the model is shown Figure 26. Notionally, there is a logical mapping from the JSON to the SysML model structure shown in Figure 27.



```json
{
    "name": "Test",
    "elements": [
        {
            "type": "LuaScriptedEntity",
            "name": "UAV",
            "linearSpeed": 5.0,
            "angularSpeed": 60.0,
            "initialLocation": { "type": "CartesianLocation", "x": 5.0, "y": 5.0 },
            "initialHeading": { "angle": 0.0 },
            "luaFile": "src/test/resources/uav.lua"
        },
        {
            "type": "LuaScriptedEntity",
            "name": "Tracker",
            "angularSpeed": 90.0,
            "initialLocation": { "type": "CartesianLocation", "x": 10.0, "y": 10.0 },
            "initialHeading": { "angle": 0.0 },
            "initialTarget": "UAV",
            "luaFile": "src/test/resources/tracker.lua"
        },
        {
            "type": "LuaScriptedEntity",
            "name": "Target",
            "initialLocation": { "type": "CartesianLocation", "x": 32.0, "y": 15.0 },
            "initialHeading": { "angle": 225.0 }
        }
    ],
    "world": {
        "type": "HexGrid",
        "data": [
            [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1],
            [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
            [1,0,0,0,0,0,0,0,0,0,0,2,2,2,0,0,0,0,1],
            [1,0,0,0,0,0,0,0,0,0,0,0,2,2,2,0,0,0,1],
            [1,0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,0,0,1],
            [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
            [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
            [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
            [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
            [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1],
            [1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1]
        ],
        "hexSize": 2.0
    }
}
```

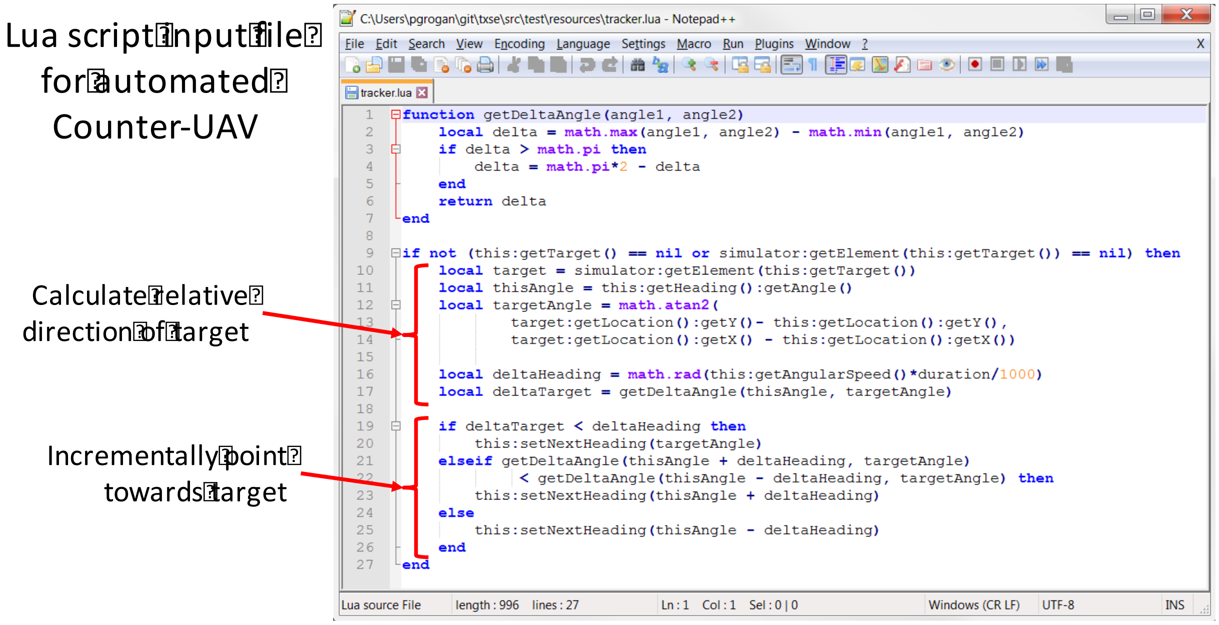
Figure 26. Mission Model – Structure



Block definition diagram (model system attributes)

Figure 27. SysML Model – Structure

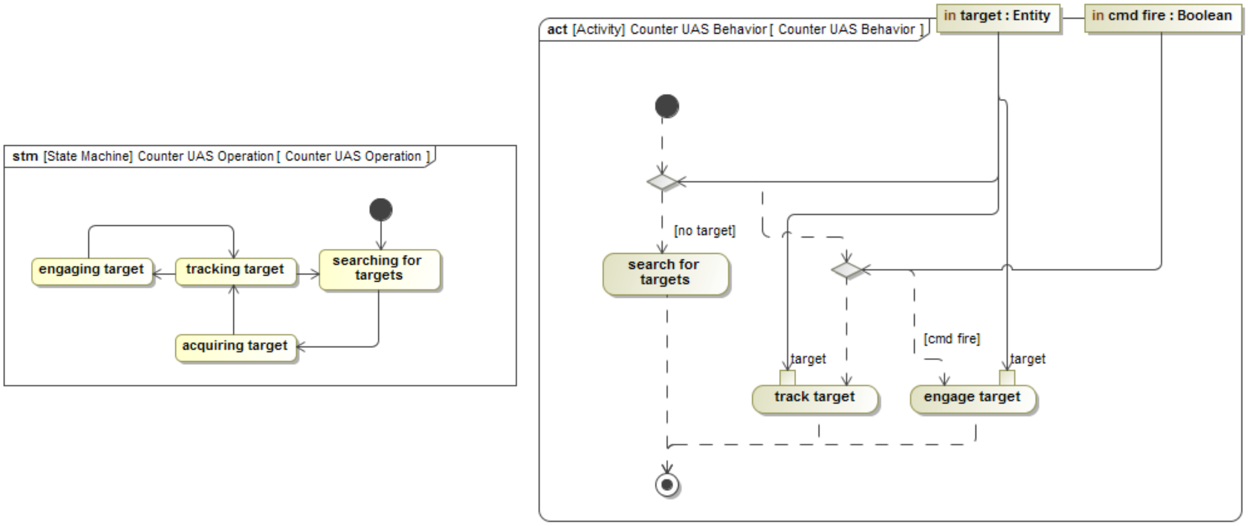
Representing behavioral information in mission modeling can be done with Lua [113] scripts as shown in Figure 28. Lua is a lightweight, embeddable scripting language (e.g., in Java). It supports procedural programming, object-oriented programming, functional programming, data-driven programming, and data description.



```lua
function getDeltaAngle(angle1, angle2)
    local delta = math.max(angle1, angle2) - math.min(angle1, angle2)
    if delta > math.pi then
        delta = math.pi*2 - delta
    end
    return delta
end

if not (this:getTarget() == nil or simulator:getElement(this:getTarget()) == nil) then
    local target = simulator:getElement(this:getTarget())
    local thisAngle = this:getHeading():getAngle()
    local targetAngle = math.atan2(
            target:getLocation():getY()- this:getLocation():getY(),
            target:getLocation():getX() - this:getLocation():getX())

    local deltaHeading = math.rad(this:getAngularSpeed()*duration/1000)
    local deltaTarget = getDeltaAngle(thisAngle, targetAngle)

    if deltaTarget < deltaHeading then
        this:setNextHeading(targetAngle)
    elseif getDeltaAngle(thisAngle + deltaHeading, targetAngle)
             < getDeltaAngle(thisAngle - deltaHeading, targetAngle) then
        this:setNextHeading(thisAngle + deltaHeading)
    else
        this:setNextHeading(thisAngle - deltaHeading)
    end
end
```

Figure 28. Mission Model of Behavior

In SysML behaviors can be represented in state machine (stm) or activity (act) diagrams as shown in Figure 29. SysML behaviors can also be represented in sequence diagrams (not shown here). While these are intuitive abstractions, the diagrams cannot easily be transformed to scripted code (e.g. Lua script), because they are usually more abstract to facilitate documentation; this could notionally double the effort to implement and completely document the models.



State machine diagram (model system states)

Activity diagram (model system actions)

Figure 29. SysML Models of Behavior

The following lists some of the challenges with the integration to SysML:

- Lack of "acceptable" representations and transformation using SysML; we are planning to investigate this more deeply in UC04
- Graphical diagrams specified at multiple abstractions
- Oriented towards concrete design
- Likely to be missing relevant mission/scenario parameters
- XMI is difficult to 'query' for structural parameters, and some of the SysML tools are moving to other technologies to address some of these limitations
- Low-level with extensive unique IDs difficult to interpret/parse
- Behavioral diagrams cannot easily be transformed to scripted code (e.g. Lua script)

The overarching challenge is the difficulty of tool-to-tool integration. This is again the reason we demonstrated a simple example using interoperability using the underlying information model with SWT, SysML, and OpenMBEE in the context of IoIF.

5.2 USING SEMANTIC WEB TECHNOLOGY FOR MISSION MODELING AND SIMULATION

In support of UC00, this use case is being extended to research the use of centralized shared information using the IoIF and specifically the use of SWT by:

- Populating the system model represented in the SWT using sensor data from other simulations.
- Query the system model (i.e., SWT, SPARQL) to retrieve specified design attributes (e.g. retrieve system attributes as inputs to the mission analysis)
- Store analysis results for later use by other modules (e.g. store mission analysis results for use in downstream decision support modules)

The current research extends the 2D modeling and simulation environments for distributed simulations to integrate through the components of the IoIF as shown in Figure 6. As discussed in

Section 4.1, we demonstrated this concept for our sponsors using the IoIF SWT. As shown in Figure 31, we created a simplified version of a use case to demonstrate data exchange, which is a subset of the functionality of the IoIF:

- UAV model: output system performance attributes
- C-UAV model: output system performance attributes
- Mission model: evaluate system performance in context of simulated mission

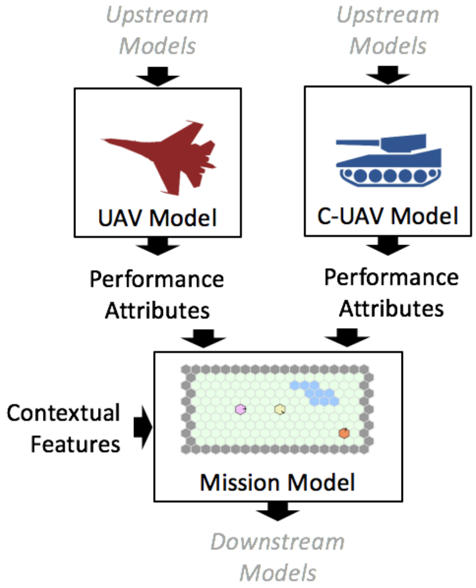


Figure 30. UC01-UC03 Prototype Application Case

We created a simple ontology, not for the purpose of illustrating how to develop a "proper ontology," but more as the basis for showing examples of using SWT for interoperability using the IoIF. The small ontology describes class of shared information using OWL, object properties, and data properties, as shown in Figure 31. The model instances corresponding to the red and blue systems are produced in RDF, and then added to a triple store. SPARQL queries retrieve and update values to create a dynamic interaction through the Data Acquisition and Aggregation layer (DAA) in conjunction with the SWT as shown in Figure 32.

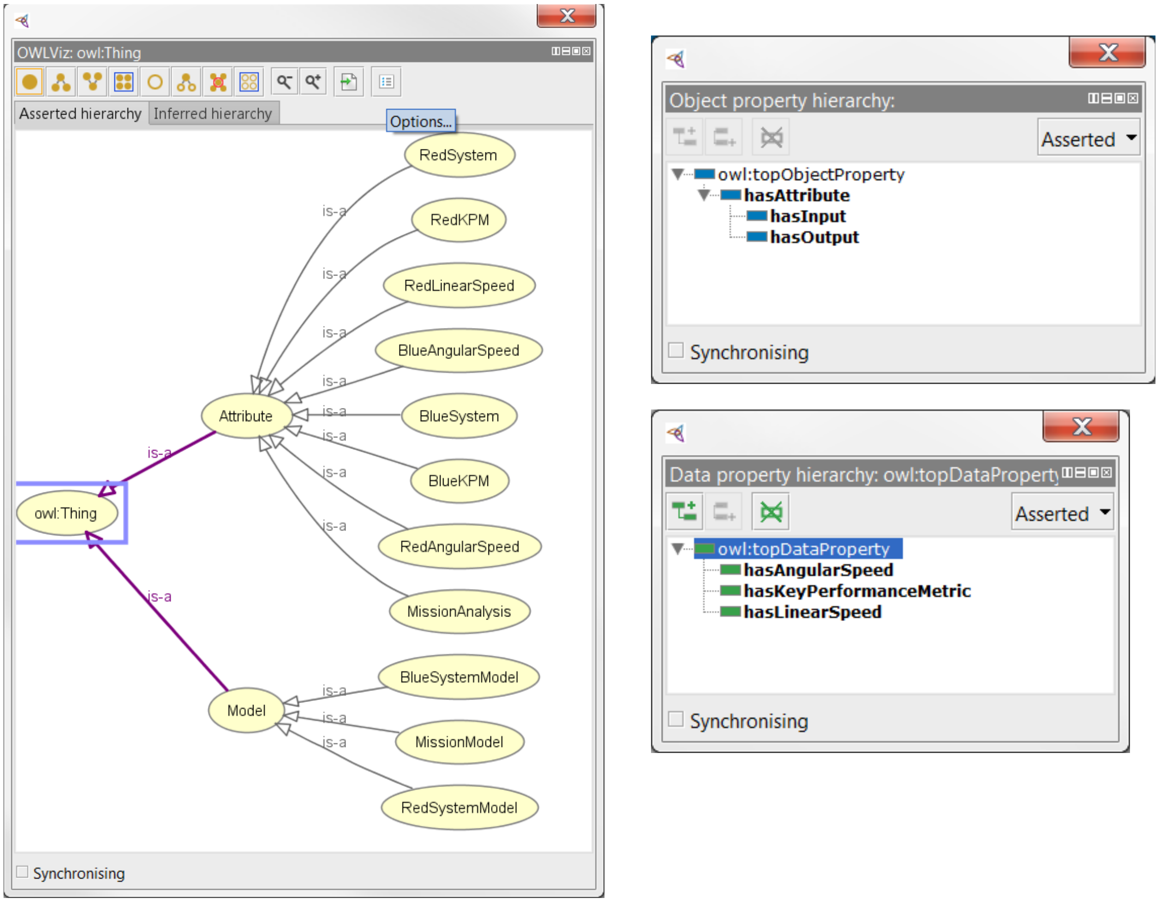


Figure 31. Simple Ontology for Experiment of Simulation Integration the SWT

Some examples of the underlying details of the information described in the ontology are shown below in the Terse RDF Triple Language (Turtle). The Subject-Predicate-Object triples are easier to read in Turtle than the underlying XML. For example ":Attribute is a rdf:type of the owl Class." In general, most user of this type of underlying technology never see this level of detail, and we refer interested readers to other sources [202].

```
:Attribute rdf:type owl:Class ;
      rdfs:subClassOf [ rdf:type owl:Restriction ;
                  owl:onProperty :hasUnits ;
                  owl:someValuesFrom xsd:string
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :hasValue ;
                  owl:someValuesFrom xsd:double
                ] .
:hasUnits rdf:type owl:DatatypeProperty ;
      rdfs:subPropertyOf owl:topDataProperty ;
      rdf:type owl:FunctionalProperty ;
      rdfs:domain :Attribute ;
      rdfs:range xsd:string .
:hasValue rdf:type owl:DatatypeProperty ;
      rdfs:subPropertyOf owl:topDataProperty ;
      rdf:type owl:FunctionalProperty ;
      rdfs:domain :Attribute ;
      rdfs:range xsd:double .

:UAV rdf:type owl:Class ;
   rdfs:subClassOf [ rdf:type owl:Restriction ;
```

```
                  owl:onProperty :hasMaxSpeed ;
                  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                  owl:onClass :MaxSpeed
                ] ,
                [ rdf:type owl:Restriction ;
                  owl:onProperty :hasTurnRate ;
                  owl:qualifiedCardinality "1"^^xsd:nonNegativeInteger ;
                  owl:onClass :TurnRate
                ] .
:MaxSpeed rdf:type owl:Class ;
        rdfs:subClassOf :Attribute .
:TurnRate rdf:type owl:Class ;
        rdfs:subClassOf :Attribute .
:hasMaxSpeed rdf:type owl:ObjectProperty ;
          rdfs:subPropertyOf :hasLinearSpeed ;
          rdf:type owl:FunctionalProperty ;
          rdfs:range :MaxSpeed .
:hasTurnRate rdf:type owl:ObjectProperty ;
          rdfs:subPropertyOf :hasAngularSpeed ;
          rdf:type owl:FunctionalProperty ;
          rdfs:range :TurnRate .
```

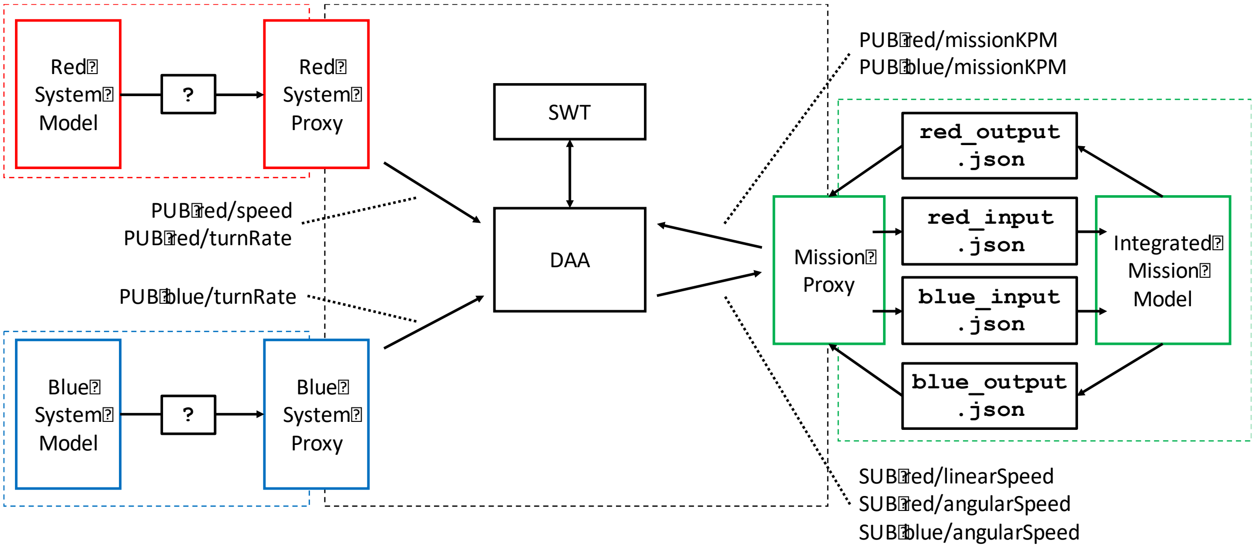
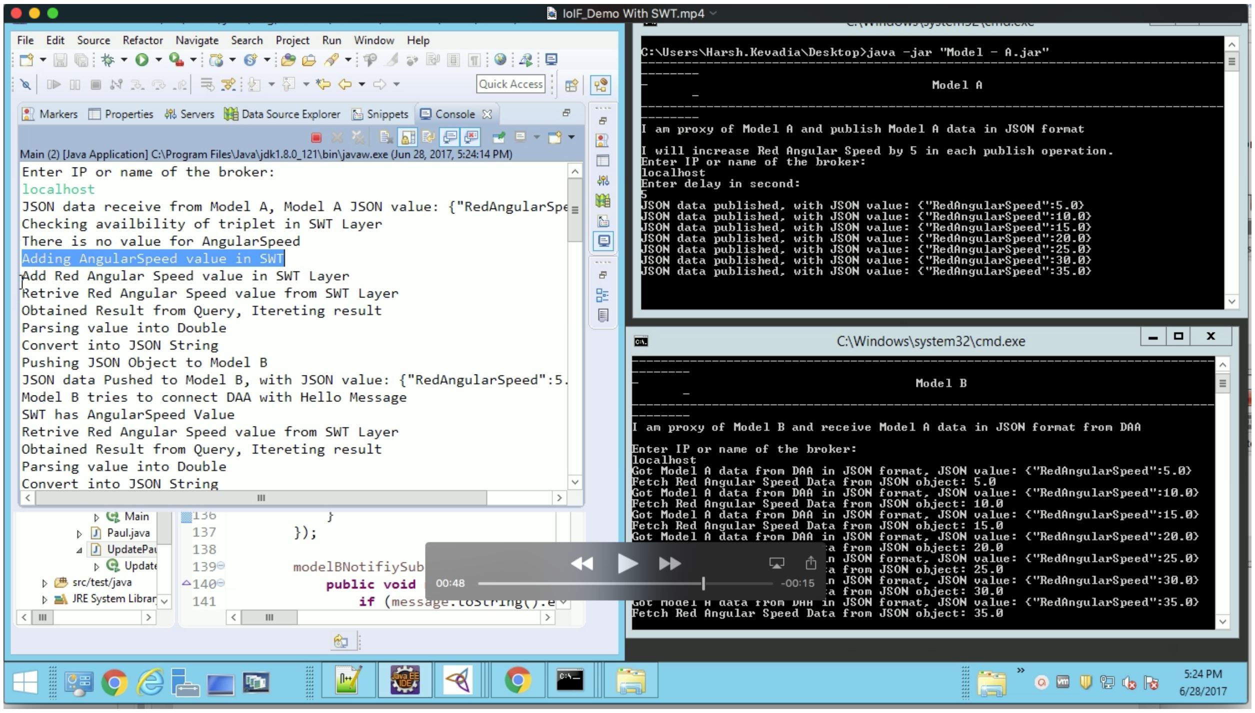


Figure 32. Multi-fidelity Mission Simulation using Semantic Web Technology and Data Acquisition and Aggregation

There is a video of a simplified version of this demonstration as shown in Figure 33. The video was shown to our ARDEC sponsors. In this simple demonstration, Model A publishes data to the DAA using its proxy, which inserts the data into the triple store using a SPARQL query (note: a SPARQL query can read or write to a triple store). Model B subscribes to the “RedAngularData.” The DAA subscribe method performs a SPARQL query to retrieve the data and send to Model B proxy.



Main (2) [Java Application] C:\Program Files\Java\jdk1.8.0_121\bin\javaw.exe (Jun 28, 2017, 5:24:14 PM)

```
Enter IP or name of the broker:
localhost
JSON data receive from Model A, Model A JSON value: {"RedAngularSpe
Checking availbility of triplet in SWT Layer
There is no value for AngularSpeed
Adding AngularSpeed value in SWT
Add Red Angular Speed value in SWT Layer
Retrive Red Angular Speed value from SWT Layer
Obtained Result from Query, Itereting result
Parsing value into Double
Convert into JSON String
Pushing JSON Object to Model B
JSON data Pushed to Model B, with JSON value: {"RedAngularSpeed":5.
Model B tries to connect DAA with Hello Message
SWT has AngularSpeed Value
Retrive Red Angular Speed value from SWT Layer
Obtained Result from Query, Itereting result
Parsing value into Double
Convert into JSON String
```

```
            }
        });

        modelBNotifiySub
            public void
                if (message.toString().e
```

```
C:\Users\Harsh.Kevadia\Desktop>java -jar "Model - A.jar"
--------------------------------------------------------------------------------
--------
-                                  Model A
       -
--------------------------------------------------------------------------------
--------
I am proxy of Model A and publish Model A data in JSON format

I will increase Red Angular Speed by 5 in each publish operation.
Enter IP or name of the broker:
localhost
Enter delay in second:
5
JSON data published, with JSON value: {"RedAngularSpeed":5.0}
JSON data published, with JSON value: {"RedAngularSpeed":10.0}
JSON data published, with JSON value: {"RedAngularSpeed":15.0}
JSON data published, with JSON value: {"RedAngularSpeed":20.0}
JSON data published, with JSON value: {"RedAngularSpeed":25.0}
JSON data published, with JSON value: {"RedAngularSpeed":30.0}
JSON data published, with JSON value: {"RedAngularSpeed":35.0}
```

```
--------------------------------------------------------------------------------
--------
-                                  Model B
       -
--------------------------------------------------------------------------------
--------
I am proxy of Model B and receive Model A data in JSON format from DAA

Enter IP or name of the broker:
localhost
Got Model A data from DAA in JSON format, JSON value: {"RedAngularSpeed":5.0}
Fetch Red Angular Speed Data from JSON object: 5.0
Got Model A data from DAA in JSON format, JSON value: {"RedAngularSpeed":10.0}
Fetch Red Angular Speed Data from JSON object: 10.0
Got Model A data from DAA in JSON format, JSON value: {"RedAngularSpeed":15.0}
Fetch Red Angular Speed Data from JSON object: 15.0
Got Model A data from DAA in JSON format, JSON value: {"RedAngularSpeed":20.0}
                        a from JSON object: 20.0
                        n JSON format, JSON value: {"RedAngularSpeed":25.0}
                        a from JSON object: 25.0
                        n JSON format, JSON value: {"RedAngularSpeed":30.0}
                        a from JSON object: 30.0
Got Model A data from DAA in JSON format, JSON value: {"RedAngularSpeed":35.0}
Fetch Red Angular Speed Data from JSON object: 35.0
```

Figure 33. Video Demonstrating Integration and Interoperability Framework

An early instantiation of the research during Phase II involved five of the researcher to execute a demonstration as reflected in Figure 34. This version of the IoIF uses two active models and passes published data through the SWT layer before delivering the data to the subscribing model. The published data that is passed into the SWT is extracted in different units and by different name. The example demonstrates the ability of the IoIF to convert both units and name, through the following steps:

- SysML model used to model Red Team linear speed
- DocGen transforms SysML model data to xml format
- Proxy A captures and transforms xml data to RDF
- Proxy A publishes red team linear speed (in m/s) to DAA
- Linear speed variable name and units will not match what is needed for Proxy B
- Mission Model Proxy B subscribes to red team linear speed
- DAA handles publish and subscribe from proxies
- SWT resolves the differences in the variable naming of Red Team linear speed and also the units
- When Proxy A (DocGen) publishes a new linear speed then the DAA initiates a request to the SWT to get the needed information for the subscribers of that data (Mission Model) and sends the updated information to the subscriber (Mission Model)
- DAA stores RDF instance data
- For the Demo, the team manually changed SysML model's linear speed and re-ran Mission Model simulation to demonstrate automated propagation of data change through system

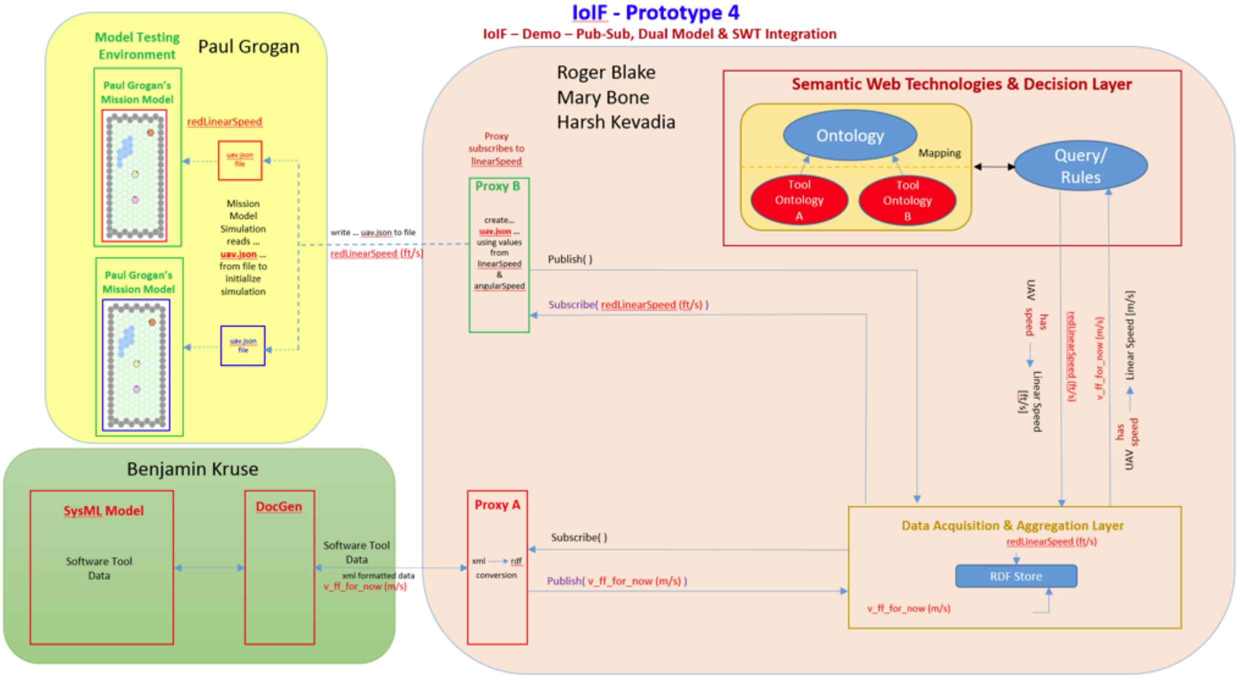


Figure 34. Integrating System Model Data through SWT to 2D Simulation

6 MULTIDISCIPLINARY, DESIGN, ANALYSIS AND OPTIMIZATION (UC03)

This use case investigates the methods to trace capabilities to the relevant design disciplines and perform cross-domain analyses through Multidisciplinary Design Analysis & Optimization (MDAO) for problem and design tradespace analyses. We also want to demonstrate the feasibility to investigate tradespace and alternatives using MDAO workflows at the CONOPS, mission, system, and subsystem levels. Some of recent research demonstrates include three different research applications developed to illustrate how our research team have used ModelCenter and MBSEPak for: 1) developing Multidisciplinary Design, Analysis and Optimization workflows for Key Performance Parameter examples at the system level, 2) ModelCenter integrated with a Graphical Concept of Operation (CONOPS) example using the Unity gaming engine at the mission level, and 3) ModelCenter and MBSEPak, with MagicDraw SysML to formalize the concept of an Assessment Flow Diagram [23]. Some of the latest demonstrations during Phase II involve applying MDAO at the CONOPS level as discussed in Section 4.2.

MDAO is an approach for calculating optimal designs and understanding design trade-offs in an environment that simultaneously considers many types of simulations, evaluations, and objectives. For example, when designing a vehicle, there is typically a trade-off between maximizing performance and maximizing efficiency, where calculating either of these objectives require multiple disciplinary models (geometry, weight, aerodynamics, propulsion). MDAO prescribes ways to integrate these models and explore the necessary trade-offs among the objectives to make a design decision. While the theoretical foundations of MDAO are well-established by academics, a number of barriers to practical implementation exist. Chief among these is the lack of model integration, which prevents designers of one subsystem from easily assessing how changing a design variable affects the results of other subsystems' models or simulations. The overarching objective of this use case is to understand these challenges and develop recommendations for overcoming them and effectively applying MDAO to add value in a large, distributed, organization such as ARDEC.

As illustrated by some of the examples in UC01 and UC02, we can extract the key parameters in these various mission and system simulations. These parameters are fundamental to the MDAO workflows. We need to combine those parameters for different elements of a workflow, but we must also characterize our key performance parameters (KPP); for example, a surveillance UAV range or endurance would be KPPs. These KPPs are modeled as the outputs from running the MDAO through different optimizations. The other aspect of the method involves identifying the constraints that must be characterized with respect to KPPs (i.e., outputs) with respect to selected inputs. As discussed in Section 9, we believe that the decision framework (see Figure 3) use case UC06 provides a methodological approach to identify the KPPs.

6.1 MDAO OBJECTIVES

More specific objectives include:

- Assessing the impacts of individual design changes on system capabilities.
- Supporting early-phase (conceptual design), system-level trade-off analysis using previous evaluation results from existing models.
- Developing strategies to transform the contracting process so that requests for proposals (RFPs) can be designed more flexibly toward value-based (rather than target-based) design; examples of this have been accomplished in the NAVAIR surrogate pilot

In pursuit of these objectives, the research activities entail:

- Develop generic multidisciplinary models of an UAS, including analyses of the geometry, structure, aerodynamics, propulsion, and performance capabilities, to be used as an example case.
- Explore using systems representations (e.g., SysML, Domain Specific Models) to map all inputs (parameters and variables) and outputs (objectives, constraints, intermediate parameters) among the individual models.
- Conduct trade studies on the UAS design using established approaches and tools for MDAO, exploring different approaches, tools, and visualization techniques to most effectively display information and uncertainty for decision-makers.
- Explore ways that previous trade study results on detail-phase product design can be useful toward new conceptual design of products with varying mission capability requirements
- Work with ARDEC project leads to understand the barriers to implementing this type of MDAO, culturally and practically/theoretically.
- Explore more general ways to map and coordinate subject matter experts (SMEs) and data, models, and meta-models for improved (1) requirements setting for Request for Proposal (RFP) or CONOPS, and (2) value-driven design.

Interfaces with other sub-tasks include:

- Explore ways to more seamlessly associate parameters from mission and system modeling and simulation for UC01 and UC02.
- Receiving and using model structures from “Use Model Based Engineering,” “Develop Information Model,” and “Create System Models” portions.

- Feeding and matching capabilities and needs with the “Research Mission and System Operational Capabilities” and “Research Graphical CONOPS” portions of the project, as well as the “Research Decision Framework” portion.
- Investigate how MDAO outputs can be further used to calibrate mission and system modeling and simulation.
- Demonstrate how MDAO can be used to formalize the Assessment Flow Diagram (AFD) for the Decision Framework (UC06); accomplished in Phase II [23].

One of the objectives of this project is to leverage the most powerful tools that are often used by industry as well as government organization. We have secured academic licenses to Phoenix Integration’s ModelCenter [156]. Further, while research to date examines the use of MDAO at the systems level. We have received additional academic licenses to ModelCenter to investigate the use of MDAO at the mission and subsystem levels.

6.2 MDAO METHODS

Using tools like ModelCenter, we have investigated, demonstrated and described methods for applying such tools, and also identifying the relevant research questions in the context of those advanced tools. For example, the steps for an MDAO method may be characterized as:

- Describe a workflow (scenarios) for a KPP (e.g., range, notionally similar to surveillance time).
- Determine relevant set of inputs and outputs (parameters).
- Illustrate how to use a Design of Experiments (DoE) and use analyses such as sensitivity analysis and visualizations to understand the key parameter to use with optimizations.
- Illustrate Optimization using solvers with 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.

A number of methods can be applied to formulate multidisciplinary optimization problems, develop useful surrogate models, and calculate optimal and Pareto-optimal solutions. Optimization problems can be formulated with a number of different objectives by converting some objectives to targets or constraints, summing the objectives with value-based and unit-consistent weighting schemes, or multiplying and dividing objectives by one another. Surrogate models are often used to quickly simulate the behavior of a more computationally-intensive simulation model, and some common methods include interpolation, response surface using regression models, artificial neural networks, kriging, and support vector machines. Finally, numerical optimization can be performed using a number of different algorithms and techniques, including gradient-based methods, pattern search methods, and population-based methods. For each of these, different techniques have been found to be more suitable to different applications, and part of this research directive will be to identify and demonstrate the best tools for this MCE architecture.

6.3 INTEGRATIONS WITH RELATED TASKS

While the theoretical foundations of MDAO are well-established by academics, a number of barriers to practical implementation exist. Chief among these is the lack of model integration, which prevents designers from easily assessing how changing one design variable affects the outputs from different models or simulations. Through this project, and the creation of an MCE architecture that follows an AST and a consistent ontology, we will be able to leverage MDAO techniques in the design decision-making process. From an academic perspective, the major contributions will be demonstration methods for integrating MDAO practices into complex existing and new organizational structures, and bringing MDAO workflows together with system engineering models.

A solid framework for MDAO can enable multi-objective optimization, showing product developers how different design objectives compete with one another. For example, we know that improving an objective like "minimize weight" typically requires a sacrifice in the objective to "maximize power." The magnitude of that improvement-sacrifice relationship, which often involves different units and requires human judgement to make a mission-appropriate decision, can be revealed by combining different simulation models, surrogate models, and optimization routines. As this may involve balancing a large number of objectives, one of the key challenges is in visualization of the results to enable informed decision-making. This fits into all five tasks of the project, as the entire information architecture must be built to support cross-disciplinary analysis, and specific tools and techniques can be integrated and tested at different stages of the transformation.

6.4 MDAO UAV EXAMPLES AND USE CASES

Demonstration covering several of the objectives have been presented in several working sessions as well as several bi-weekly status meetings. The demonstrated workflow shown in Figure 35 was developed using ModelCenter, or in conjunction with SysML and the MBSE Analyzer that provides an integration from MagicDraw SysML models to ModelCenter. This section provides a summary of the evolving use of MDAO and different workflows for four new use cases during Phase II:

1. Developing MDAO workflows for KPP examples at system level.
2. ModelCenter integrated with a Graphical Concept of Operation (CONOPS) example using Unity gaming engine at the mission level (several versions).
3. Integrating MagicDraw SysML models with ModelCenter and MBSEPak for an underwater supercavitating[2] modeling system.
4. ModelCenter and MBSEPak, with MagicDraw SysML to formalize the concept of an Assessment Flow Diagram, which is part of the Decision Framework and process [49].
5. Update to and MDAO workflow that was initially only a Computational Fluid Dynamics (CFD) solver, but now combined with Finite Element Analysis (FEA).

6.4.1 MDAO EXAMPLE FOR FIXED WING UAV

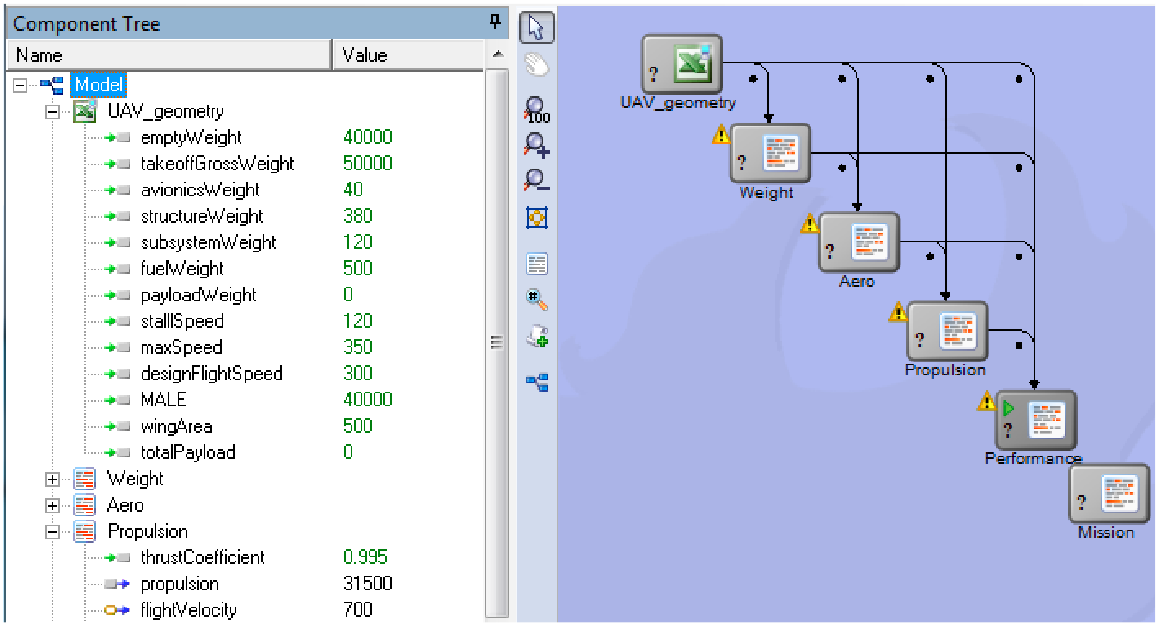
The first demonstration workflow shown in Figure 35 covered several aspects of the objectives discussed in this section, including:

- Describe and execute a workflow analysis of UAS capabilities (e.g., range, velocity, and fuel consumption).

---

[2] **Supercavitation** is the use of cavitation effects to create a bubble of gas or vapor large enough to encompass an object travelling through a liquid, greatly reducing the skin friction drag on the object and enabling high speeds

- Map relationships among parameters (inputs/outputs) in disciplinary models.
- Illustrate use of Design of Experiments (DoE), sensitivity analysis, and visualizations to understand capability relationships/trade-offs.
- Optimize using different solvers to find sets of Pareto-optimal solutions.
- Take advantage of previous model analyses for use in early-phase design with new mission capability requirements.



Component Tree

Name	Value
Model	
UAV_geometry	
emptyWeight	40000
takeoffGrossWeight	50000
avionicsWeight	40
structureWeight	380
subsystemWeight	120
fuelWeight	500
payloadWeight	0
stallSpeed	120
maxSpeed	350
designFlightSpeed	300
MALE	40000
wingArea	500
totalPayload	0
Weight	
Aero	
Propulsion	
thrustCoefficient	0.995
propulsion	31500
flightVelocity	700

Figure 35. MDAO Example Workflow

As shown in Figure 36, the Pareto frontier (Pareto optimal set) shows the trade-off between range and propulsion. The blue points show the Pareto frontier/non-dominated solutions. The Pareto frontier was calculated using a bi-objective optimization using NSGA-II algorithm to:

- Maximize range
- Maximize propulsion
- Given 5 design variables
    - Wing area (ft2)
    - Wing span (ft)
    - Altitude (ft)
    - Speed (knots)
    - Efficiency factor

These results reflect on how much range one would have to give up in order to increase the propulsion by some amount. Based on the current set of equations characterized in the workflow, the sensitivity analysis shown in Figure 37 indicates that the wing area is the variable that exhibits the clearest trade-off. The wing span has the largest effect on range, but does not present a trade-off between these objectives.

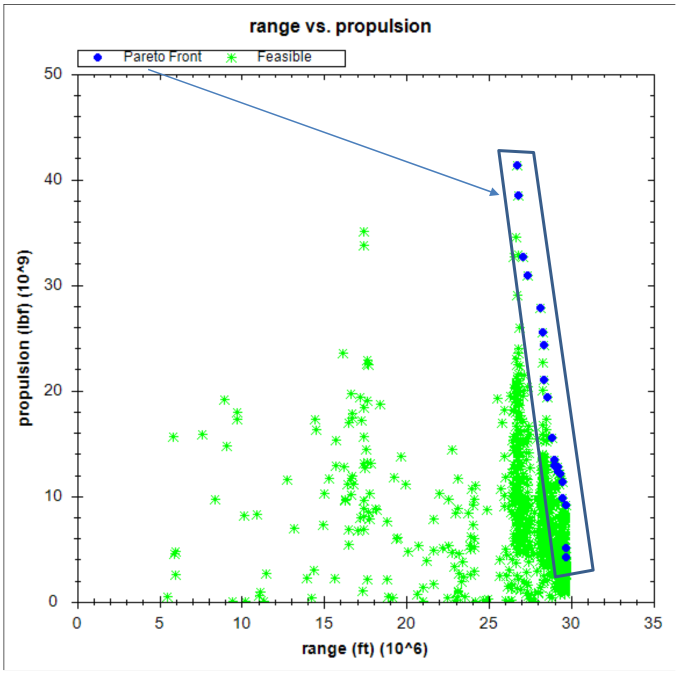


Figure 36. Pareto frontier (Pareto optimal set) Shows Trade-off Between Range and Propulsion

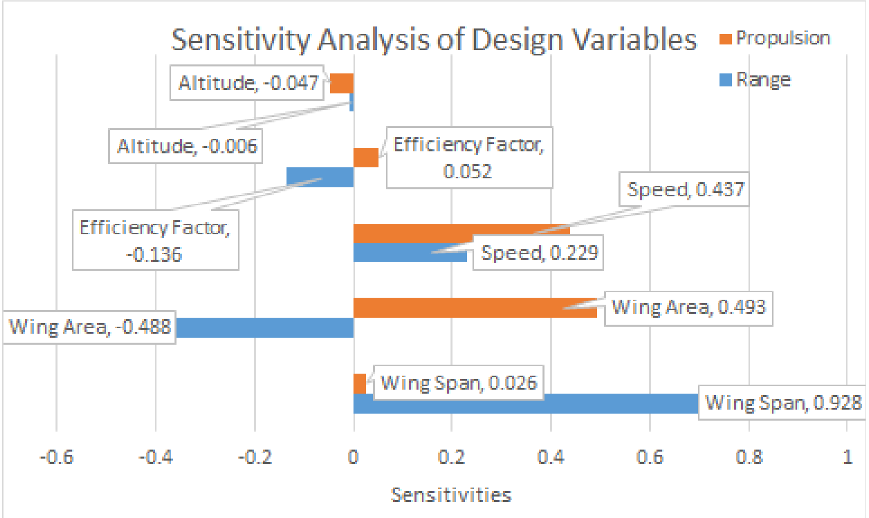
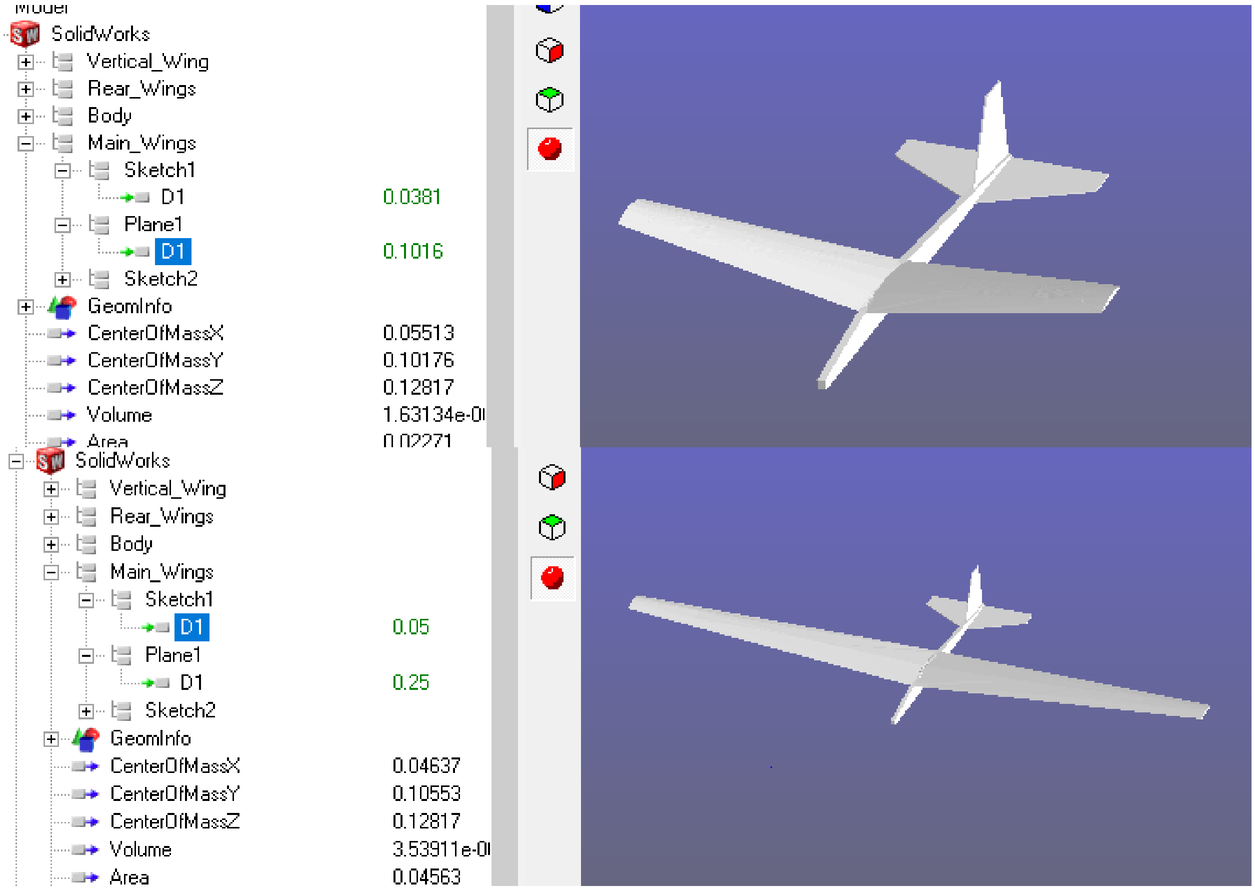


Figure 37. Sensitivity of Objectives to Design Variables

6.4.2 EXTENDING MULTI-PHYSICS MDAO UAV EXAMPLES

Brian Chell is a Ph.D. student working with Steven Hoffenson. Brian has produced a number of updates to the initial model. The efforts produced alternative workflows that leverage other types of solvers for different aspects of the problem including multi-physics problems. For example, one of the first steps looked at bring SolidWorks [184] into ModelCenter as shown in Figure 38. This provides a way to bring in detailed geometries into the analysis.



```
SolidWorks
  Vertical_Wing
  Rear_Wings
  Body
  Main_Wings
    Sketch1
      D1                0.0381
    Plane1
      D1                0.1016
    Sketch2
  GeomInfo
  CenterOfMassX         0.05513
  CenterOfMassY         0.10176
  CenterOfMassZ         0.12817
  Volume                1.63134e-0
  Area                  0.02271
SolidWorks
  Vertical_Wing
  Rear_Wings
  Body
  Main_Wings
    Sketch1
      D1                0.05
    Plane1
      D1                0.25
    Sketch2
  GeomInfo
  CenterOfMassX         0.04637
  CenterOfMassY         0.10553
  CenterOfMassZ         0.12817
  Volume                3.53911e-0
  Area                  0.04563
```

Figure 38. MDAO Workflow with SolidWorks Computer Aided Design Model

There were a few challenges with the more complicated geometries, as well as:

- Open-source geometry validity is questionable
- Model variables
  - Most SolidWorks files found so far do not import variables into ModelCenter automatically.
  - We assume we are able to set the variables within SolidWorks, but this might be more difficult because manually setting values may not align structures (e.g., wing connect to fuselage to meeting correct).
- More complex
  - Computations solver (e.g., CFD) take longer to run on the laptops provide to students.

This has led to the following investigations:

- Equation-based models derived from the model shown in Section 6.4
  - Uses publically available data on Unmanned Combat Air Vehicles (UCAV) [112] parameters
  - Model is fully operational
  - Based on weight fractions that are more scalable, and easier to change than UCAV model
  - Model starting with payload weight vs. range vs. endurance tradeoffs
  - Looking at the potential to merge with future Computational Fluid Dynamic (CFD) results
- Simulation-based models
  - Difficulties

    - Still problems with importing variables into ModelCenter
    - Very large number of variables automatically imported (12,000+)
    - Under construction
  - OpenVSP [151] vs. Solidworks (CFD)
    - OpenVSP is a parametric aircraft geometry tool
    - OpenVSP allows the user to create a 3D model of an aircraft defined by common engineering parameters. This model can be processed into formats suitable for engineering analysis.
    - OpenVSP commonly used with ModelCenter
    - SolidWorks has stronger analysis capabilities
    - OpenVSP is limited to a standardized shape library
    - SolidWorks Flow Simulation can handle turbulence
    - OpenVSP CFD is most valid at nominal flight conditions (e.g. low angle of attack)
    - OpenVSP should be sufficient for conceptual design phase

OpenVSP is being used for CFD. It is easier to use with limited library of shapes of quadcopters and fixed wing, and can run ‘headless’ (i.e., without GUI) to make computations less expensive. NASA has been using this with ModelCenter. The initial model:

- Integrated parametric geometry and CFD into ModelCenter
- Performing optimization and DOE to characterize model
- Trying to find lowest-fidelity mesh that produces accurate results
- Challenges:
  - Takes some time to change between different aircraft
  - Future NASA wrapper will make this much easier
  - High-fidelity CFD simulations are very slow; we know it can run much faster, because we tested on a MacBook Pro computer; we have not tried it on the server, because we don’t have enough licenses

Figure 39 show the CFD results from the same geometry under the same flight conditions with different fidelity meshes. The simulation on the left has a coefficient of lift many magnitudes higher than the one on the right. The next steps will:

- Investigate mesh balancing accurate results and low computing cost
- Start integrating structural analysis
  - First use built-in OpenVSP outputs for wings modeled as simple beams
  - Investigate using Finite Element Analysis (FEA)
  - While this is using an airplane in the example, the concept is relevant to things that ARDEC designs that must fly (e.g., quadcopters)

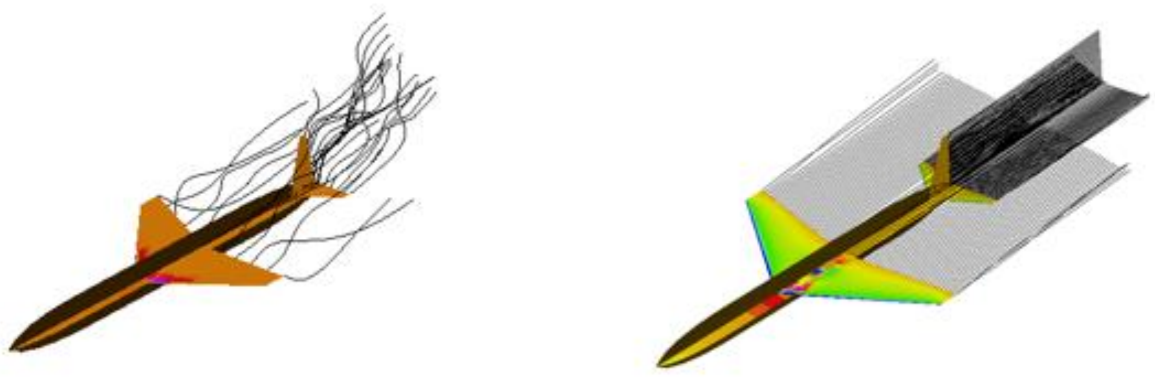


Figure 39. CFD Mesh Fidelity Importance

More recent updates include analysis for both CFD and FEA with the objective to maximize endurance and range, and minimize stress at every span-wise node. This is done with a new workflow as shown in Figure 40, with the resulting aircraft shown in Figure 41.

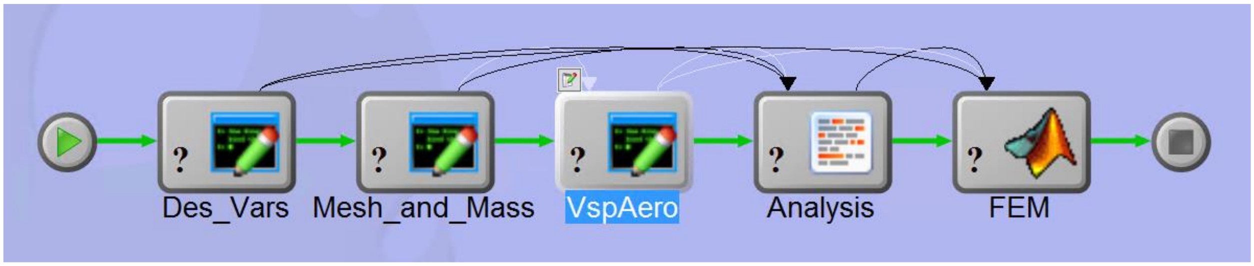
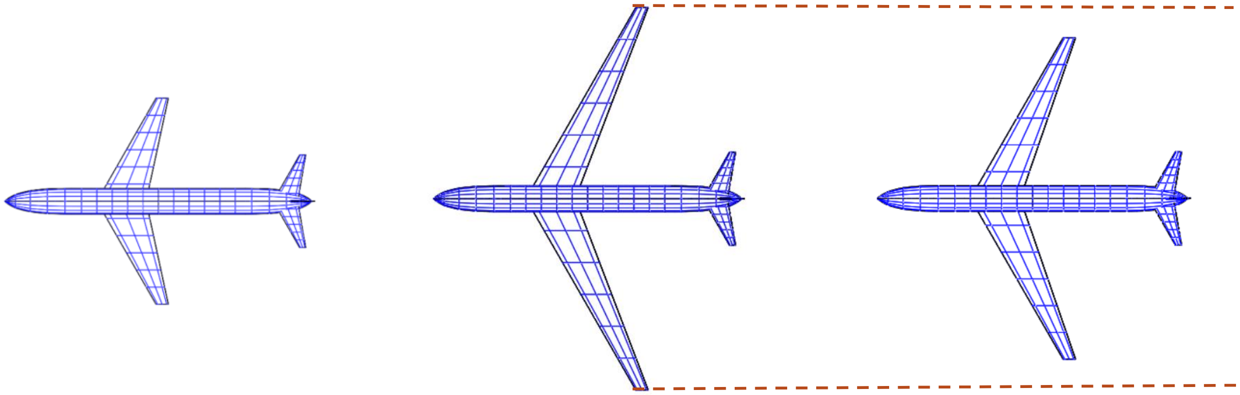


Figure 40. Update MDAO Workflow including CFD and FEA



Initial Inputs

Optimization without FEA

Optimization with FEA

Figure 41. Resulting Aircraft Designs with and without FEA

6.5 MDAO AT THE MISSION LEVEL USING GRAPHICAL CONOPS

The use case that investigated an extension of the prior work to using the Graphical CONOPS technologies Unity gaming engine with MDAO using ModelCenter is discussed in Section 4.2. The MDAO methods used:

- Design of Experiments (DoE) to run the simulation over the entire range of every input variable
    - Choose an appropriate DoE sampling method to shorten run time
        - Full Factorial
        - Latin Hypercube
- Sensitivity Analysis
    - Find which outputs are most sensitive to which input variables
    - Can remove (or fix the value) of non-sensitive variables to save time during optimizations
- Optimization
    - Use algorithm to optimize desired objective(s)

While there were challenges that were overcome, the experiment demonstrated that it is possible to use MDAO to optimize for mission success, and the number of experiments (runs) to cover the DoE space of 1000s cases versus 10s of cases that would be covered by running the scenarios manually.

The finding suggests that MDAO can be used to optimize for system-level mission success to study far more trades than can be performed manually. The initial attempt created the simulation and removed the CONOPs visualization using a "headless" simulation that is wrapped by ModelCenter. Initially the architecture of the simulation was not enabled to operate in batch modes, and therefore the software had to be re-written to work with ModelCenter. When the simulation is running, the human cannot make edits, but the re-written and wrapped simulation can run thousands of design of experiments (DoE). The initial simulation ran in real-time, but a recent update now can run faster than real-time.

A time-step analysis for the new design that can run faster than real-time for the current three missions suggests:

1. The time-step is very dependent on the timescales and complexity of the mission.
2. We have initial measure of the quality of a run through the measurement of the average kinetic energy of the blue drone. The simulation fails when this number suddenly drops.
3. For our current missions, we can increase the physics time step by about a factor of two (2). This means a speedup of about a factor of two (2). This is a smaller number than was expected going into the study.
4. In physics simulations, one is often not interested in high-frequency behavior because we are interested in long-term bulk behavior of matter. In our case, we are very interested in high-frequency behavior because that behavior is used to determine the response of the agents to each other. This is like a basketball game in which players have head fakes and tells on short timescales. The defense must pick up on the short timescale events in order to respond on longer timescales.

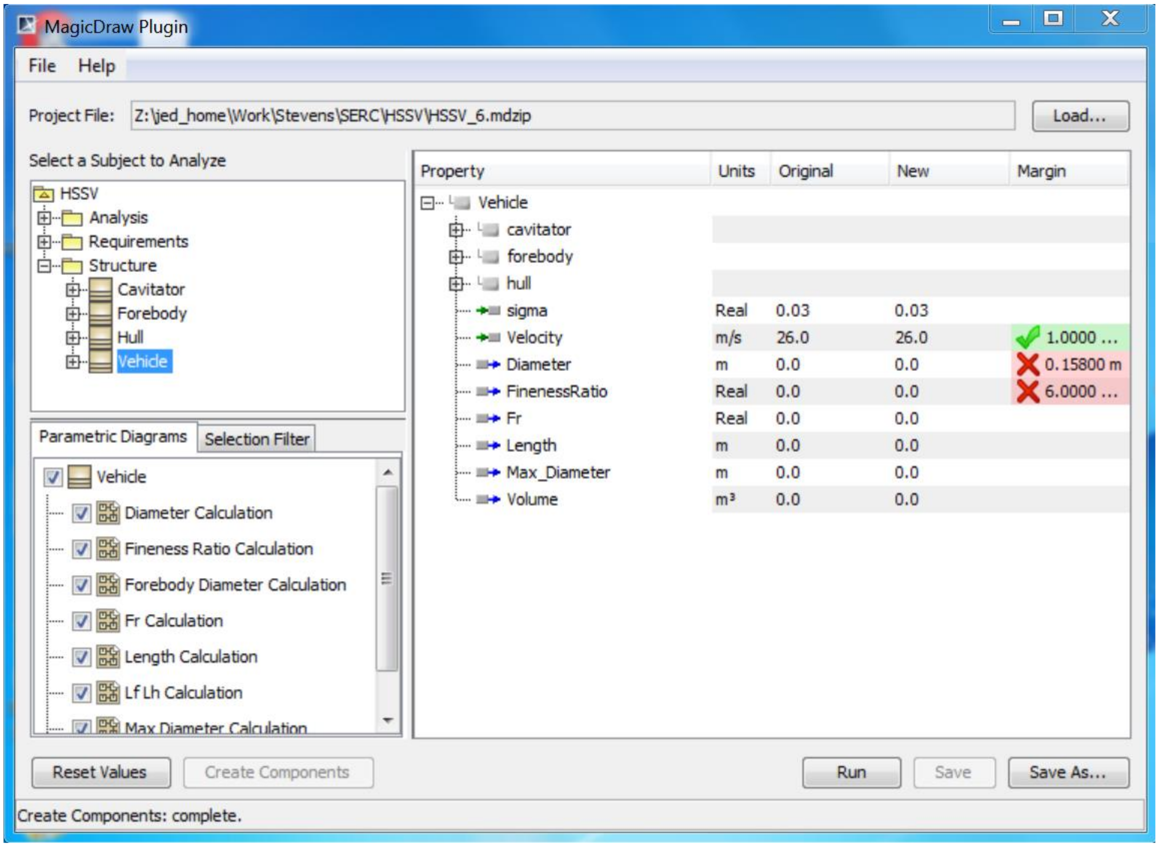
The proposed next steps include:

1. Check the sensitivity of the speed-up with other missions. The current red drone strategy is to pursue the blue drone and interfere with it.
    - What if the red strategy was more of a zone defense?
    - What emergent surprises will we see?
    - We saw unexpected flocking behavior in the current red strategy?
2. This red strategy is very different than the one we currently have. How does this affect the speed-up?
3. Find a better measure of performance than the average kinetic energy.
4. Incorporate this into the output file so that it is convenient analysis and programmatic processing.

6.6 SYSML INTEGRATION TO MDAO THROUGH MBSEPAK

This research investigated the use of the Phoenix Integration MBSEPak (formerly known as the MBSE Analyzer) that provides a way to integrate MagicDraw SysML models with ModelCenter for performing MDAO analysis [23]. John Dzielski who performed this research primarily works in MATLAB, and he used an example that was familiar to him related to underwater super cavitation modeling. The process covered the following steps:

- Defining requirements models in SysML
    - MBSEPak works by adding a profile that includes a number of stereotypes to MagicDraw
    - Specify a constraint (='s), upper and/or lower bounds, and units
- Properties are connected to requirements via the satisfy relationship
- Information is transferred to the ModelCenter through MBSE Analyzer plugin as shown in Figure 42
    - Requirements are shown in the Margin column of the plug-in.
    - The plug-in indicates whether the requirements are satisfied or not by a design
- MagicDraw Plug-In populates an analysis to create a workflow
    - Components correspond to constraint blocks
    - Constraints blocks are models or equations used in par diagrams
    - Constraint parameters correspond to component variables in ModelCenter
- Parametric (PAR) blocks are used to indicate to ModelCenter how to connect component I/O (values) to model values
- All of the other types of analyses discussed previous can then be applied in ModelCenter



Project File: Z:\jed_home\Work\Stevens\SERC\HSSV\HSSV_6.mdzip

Property	Units	Original	New	Margin
Vehicle				
cavitator				
forebody				
hull				
sigma	Real	0.03	0.03	
Velocity	m/s	26.0	26.0	✓ 1.0000 ...
Diameter	m	0.0	0.0	✗ 0.15800 m
FinenessRatio	Real	0.0	0.0	✗ 6.0000 ...
Fr	Real	0.0	0.0	
Length	m	0.0	0.0	
Max_Diameter	m	0.0	0.0	
Volume	m³	0.0	0.0	

Create Components: complete.

Figure 42. Example of MBSE Analyzer MagicDraw Plugin to Integrate with ModelCenter

6.7 FORMALIZING ASSESSMENT FLOW DIAGRAMS AS MDAO WORKFLOW

For populating the Decision Framework [49] as discussed in Section 9, we need to collect all of the elements of information. The research objective is to determine how/where to collect all of the information reflected Figure 44 from rigorously specified models. Based on inputs from Dr. Matt Cilli, some of the underlying computations are going to be published in a journal paper. This would allow us to perform most of the computation directly on the data stored in a triple store, and then extract information directly for the visualization. Matt is using this approach with the research affiliated with the Engineered Resilient Systems effort and created the visualization using Tableau software. This would provide senior leaders and program managers the type of information they need to consider technology capability tradeoff using Performance, Cost (Affordability), Time (delivery schedule) and Risk, as shown in Figure 43.

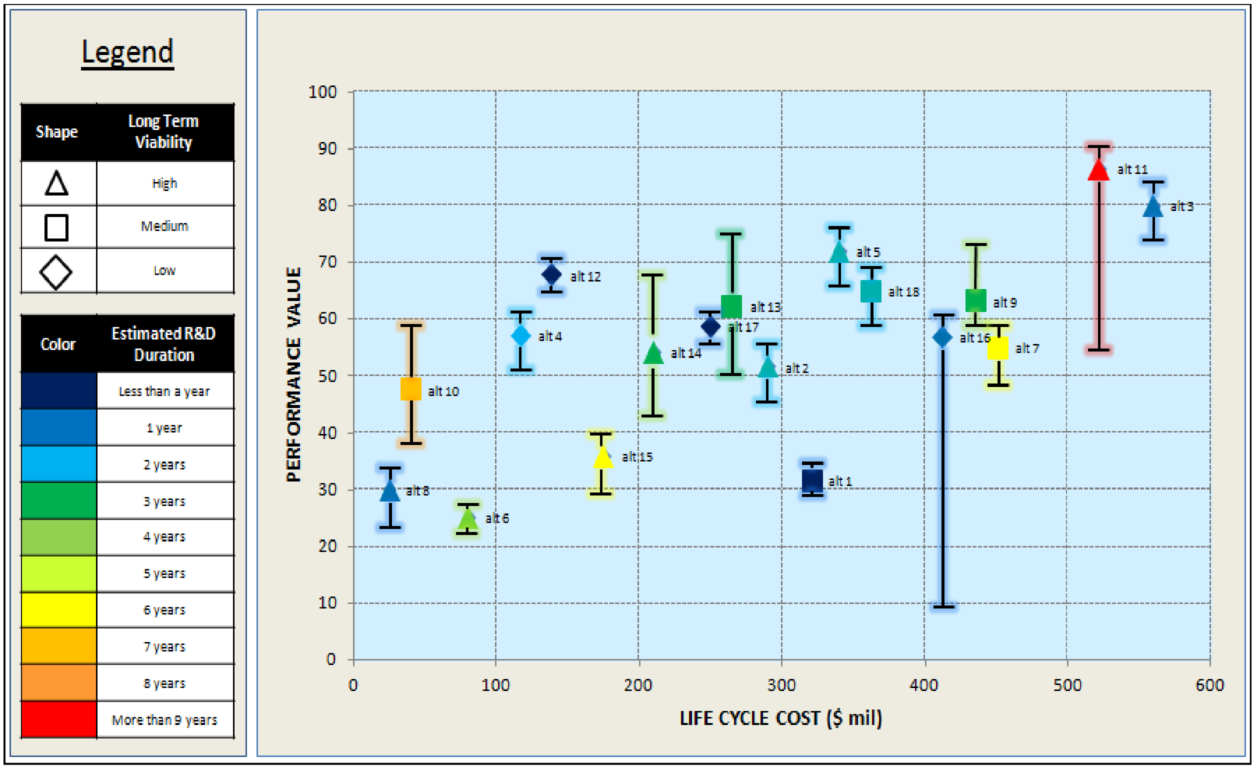


Figure 43. Visualizing Alternatives – Value Scatterplot with Assessing Impact of Uncertainty

Fundamentally, if a particular answer was unacceptable, using the concept discussed herein, we could trace linkages through the Information model back to all other related perspectives on the system in terms of operational, mission, system, and subsystem design alternatives and trades. These elements would include:

- Objective hierarchies – goals for decision making
- Value functions
- Assessment Flow Diagrams (AFDs) trace the relationships between physical means, intermediate measures, and fundamental objectives
- Uncertainties

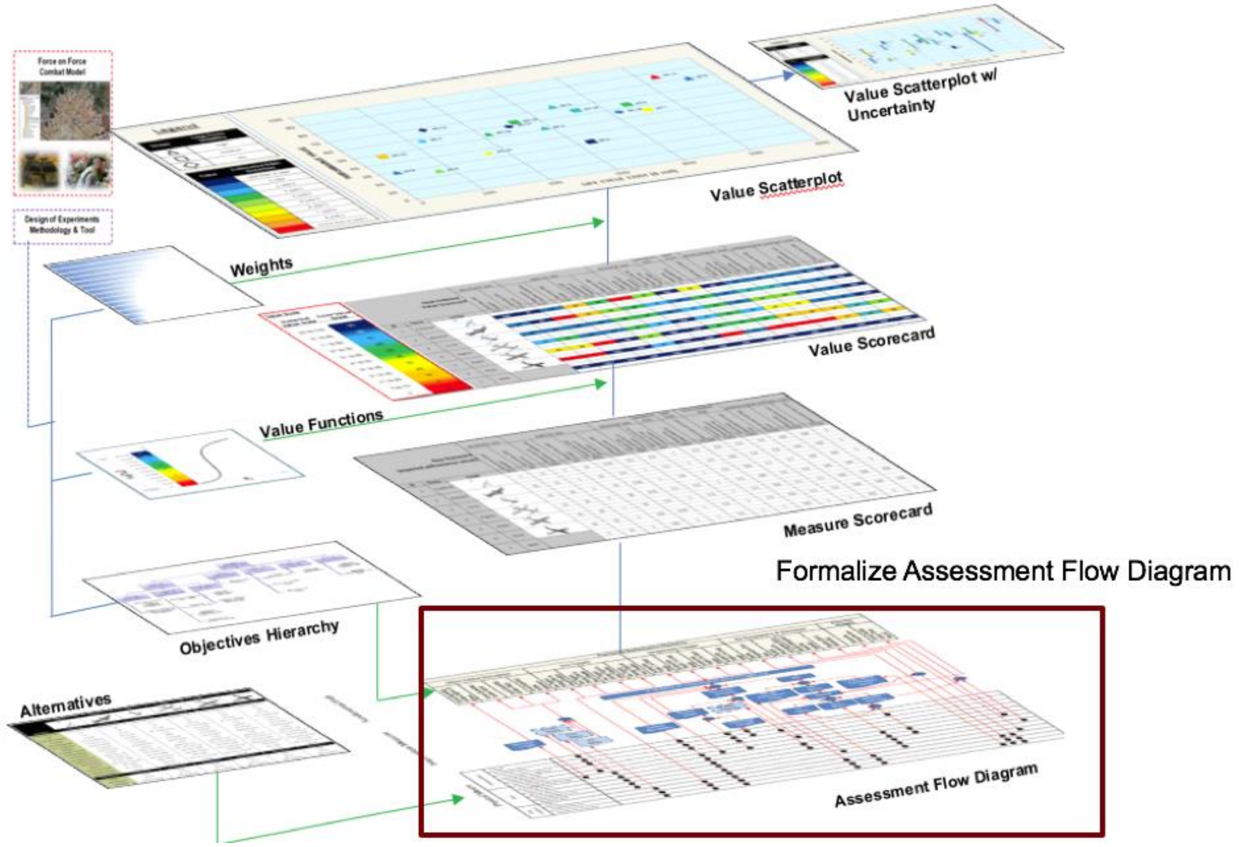


Figure 44. Decision Support Model Construct

This research used a case study documented in Matt Cilli's book chapter. We focused on formalizing the AFD using SysML, which is usually done in PowerPoint, as shown in Figure 45. This research demonstrated that we can formalize the AFD in SysML and be transformed into an MDAO workflow [23]. John started with SysML and used the MBSEPak to produce the MDAO workflow, as reflected in Figure 45, which provides a basic conceptualization for researching this concept and to address the questions:

- Can MDAO represent Assessment Flow Diagram?
- Does AFD characterize needed MDAO workflows?

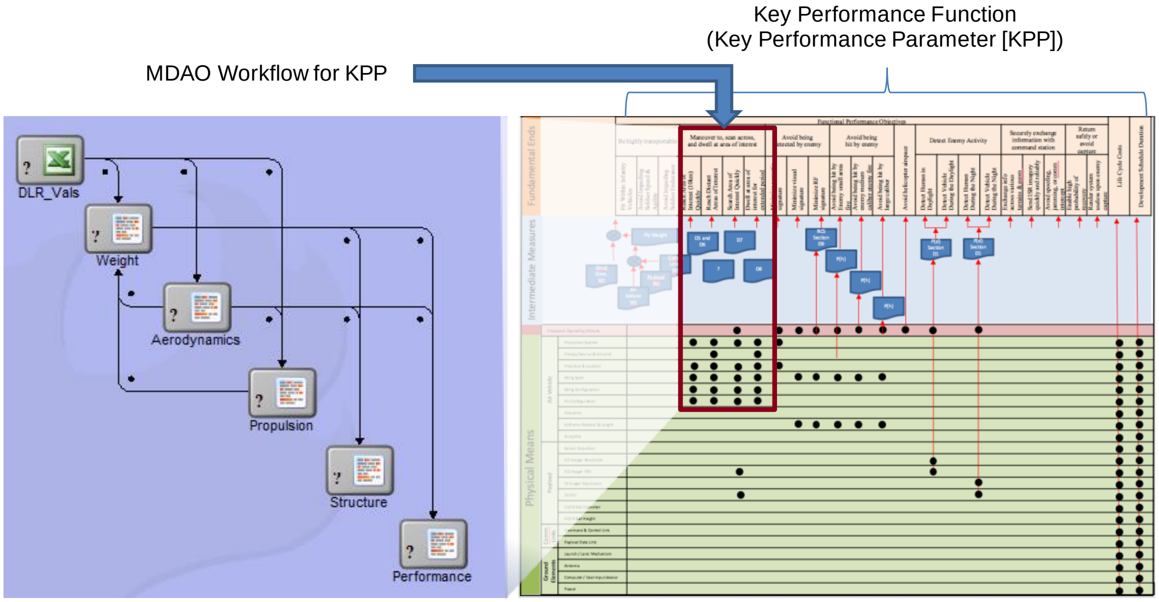
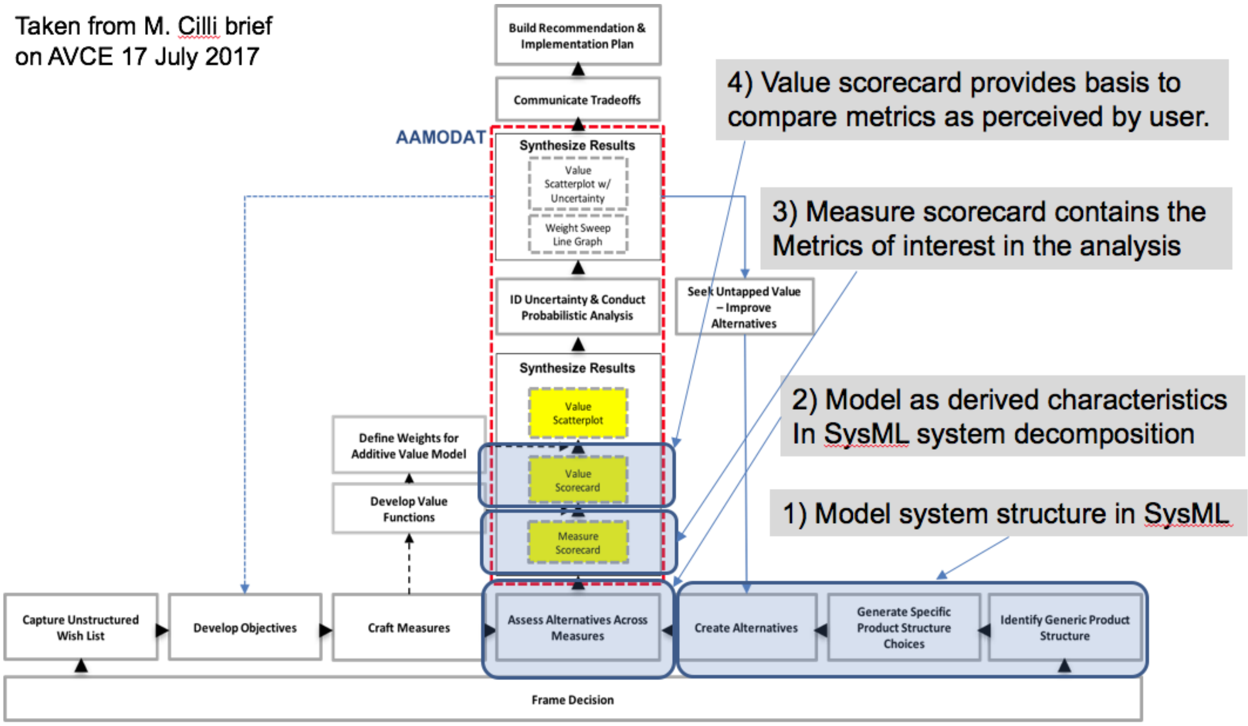


Figure 45. Formalizing the Assessment Flow Diagram

The current results have formalized the representations of AFD using SysML, MBSEPak and ModelCenter, because the Key Performance Parameters can be mapped to one or more MDAO workflows as reflected in Figure 45. With some recommendation on modeling best practices for using MBSEPak with SysML from Phoenix Integration. A Webinar explaining this approach is provided at the Phoenix Integration website (https://www.phoenix-int.com/learn-more/webinars/) called “Applications for Three Research Use Cases in Model Centric Engineering using ModelCenter and MBSEPak.” [156]

The modeling steps follow from the Decision Support Construct:

1. Model system structure in SysML
2. Model as derived value types in SysML decomposition
3. Add the needed Measure scorecard that contains the Metrics of interest in the analysis
4. Value scorecard provides basis to compare metrics as perceived by user



Taken from M. Cilli brief on AVCE 17 July 2017

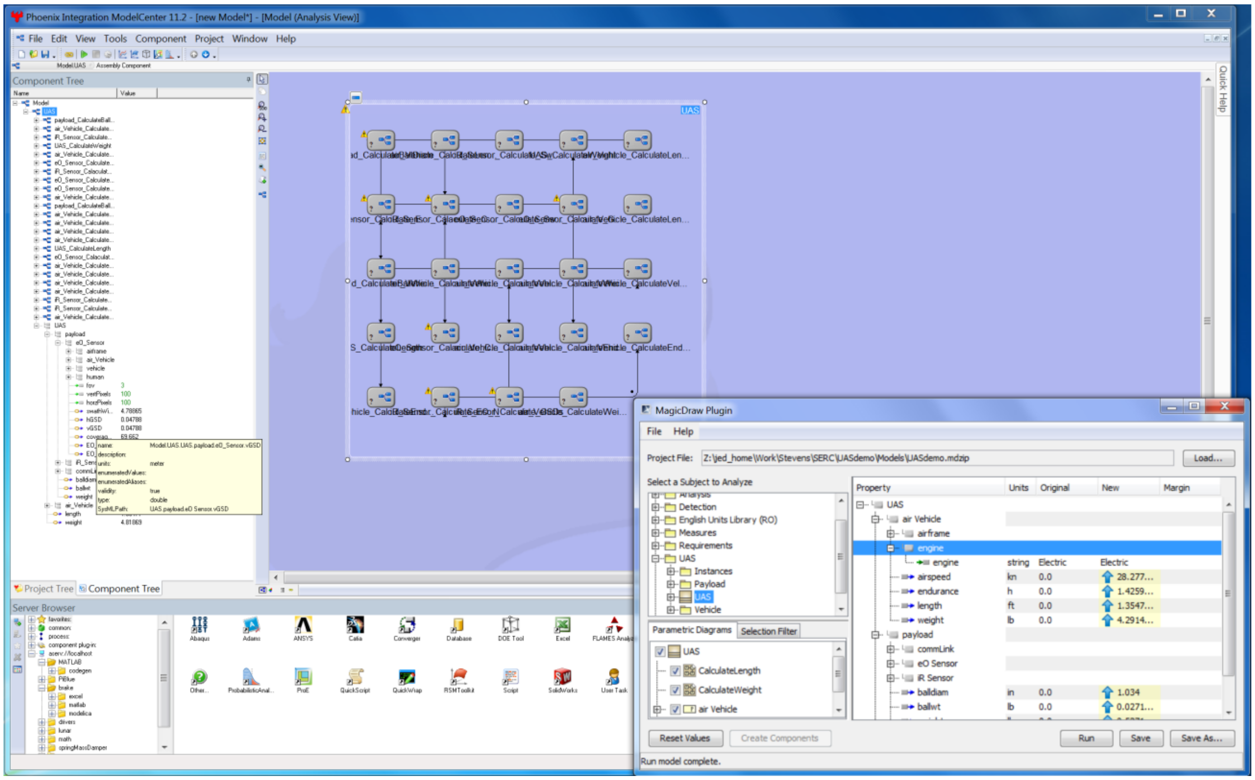
4) Value scorecard provides basis to compare metrics as perceived by user.

3) Measure scorecard contains the Metrics of interest in the analysis

2) Model as derived characteristics In SysML system decomposition

1) Model system structure in SysML

Figure 46. Decision Support Model Construct



Project File: Z:\jed_home\Work\Stevens\SERC\UASdemo\Models\UASdemo.mdzip

Run model complete.

Figure 47. MBSEPak Creates Analysis Workflow and Checks Data Type Consistency

6.8 FUTURE RESEARCH FOR MDAO

6.8.1 EXTENDING MULTIPHYSICS MDAO WORKFLOW

In an effort to improve on the research discussed in Section 6.4.2, the fidelity of ModelCenter's process of mission selection for drones such as the MQ-9 Reaper, Global Hawk, and/or Predator, additional detail of the segmented phases of a single reconnaissance mission scenario are required. The initial wing loading study used MATLAB scripts to evaluate the wing loading (weight at take-off w.r.t. span length) and power loading (propulsion w.r.t. weight at take-off) and provides more detailed insight into the following mission phases: (i) take-off/landing, (ii) cruise / loiter / maximum range, (iii) turning maneuvers.

Similar to the system's engineering MDAO approach by Becar [13], the fundamental equations and assumptions for the various mission phases are referenced from the following authors [42] [118] [119] [120]. Assumptions such as the following were made:

- Minimum velocities during take-off, cruise, and landing are assumed to be 20 percent greater than the stall speed.
- During the turn maneuvers, the quantity of turns and loading factor, "n", is used to scale/re-adjust the velocity throughout the turn that is reflected on the wing-loading and power-loading output.
- During the loiter/cruise/maximum range segment, the goal is the determine the lift/drag at the current dynamic pressure, with respect to the aircraft altitude, take-off weight, and wing span.

Thereafter, the next phase of the research will be to model one of the three proposed military drones into OpenVSP such that the VSPAero can perform a relatively quick CFD lattice analysis. Although the lattice methodology uses 2-D plates and is less insightful than the 3-D CFD analysis, the qualitative performance will be even across the board.

6.8.2 RE-PARAMETERIZATION TO MDAO FROM HIGH FIDELITY MODELS

At the request of David Allsop from Boeing, we also connected a few people from our NAVAIR visits to discuss the issue of deriving MDAO parametrics from high-fidelity models, or more generally having some type of bi-directionality between parametric models and higher fidelity simulations (which can “break” the parametric chains). Dr. Dave McCormick who runs the MDAO lab for Northrop Grumman gave an informative presentation at the April NDIA Modeling and Simulation bi-monthly committee meeting on some of challenges, which we believe are relevant to future research, such as:

- Rapid re-parameterization of completely new concepts
- Ability to incorporate static models
- Ability to bring in static changes “underneath” the parameterization
- Ability to incrementally add to parameterization
- Ability to rapidly alter the sizing logic behind models

7 SYSTEM MODELS AND MODEL BASED SYSTEMS ENGINEERING (UC04)

This use case applies MBSE methods and tools to the case study examples and also looks at how metamodels or metadata is represented in the Information Model (UC00) to provide traceability through the other forms of modeling for UC01, UC02, UC03, UC05, UC06 and UC10. This use case is developing different variants of UAS system models at both the system and mission level. We are also interested in applying MBSE methods using SysML with MagicDraw [143] to investigate benefits and synergies through OpenMBEE [150], as discussed in Section 7.1. We used the Model Development Kit (MDK)/DocGen to generate visualizations of the requirements from the AVCE iMBE model provide by the ARDEC sponsors. The use of MagicDraw also allows for integration to ModelCenter through MBSE Analyzer, as a means for modeling system constraints in SysML and integrating with MDAO as discussed in Section 6.5.

7.1 OPENMBEE AND MODEL DEVELOPMENT KIT

Most working and virtual sessions conducted with our sponsor use SysML with OpenMBEE. OpenMBEE has been evolving over the years, and we are part of the leadership team in the OpenMBEE collaboration group (https://groups.google.com/d/forum/openmbee/), which has about 230 group members, including industry participation from Boeing, Lockheed and international organizations. We believe it will be an effective tool and community for our research, but can also provide us with insights that might be beneficial to AVCE iMBE.

As shown in Figure 48, OpenMBEE has three main components: MDK – with DocGen, Model Management System (MMS), and the View Editor. DocGen works from a View and Viewpoint hierarchy, which is a type of model embedded within a system model. In the absences of more rigorous checking such as the NASA/JPL ontologies [102], or validation rules from in MagicDraw, the use of the View and Viewpoint hierarchies can be used to enforce some methodological guidelines. For example, after generating a document using DocGen, blank sections reflect potential incompleteness in the model. While the generated documents can provide a type of specification, they are often used first as a means of checking the view of a model and then “pushed” into the MMS where they can be viewed through the View Editor, which runs in a standard browser. The View Editor allows:

- Access by person, roles, supporting review
- Can update information that can be pushed back into the model through the MMS

NASA/JPL hoped that the process of open sourcing OpenMBEE would encourage tool vendors to add capability into the commercial tools, and to some extent this has occurred. The updates created by NASA/JPL improve the practice of modeling. Details are provided on Github: *https://open-mbee.github.io/*.

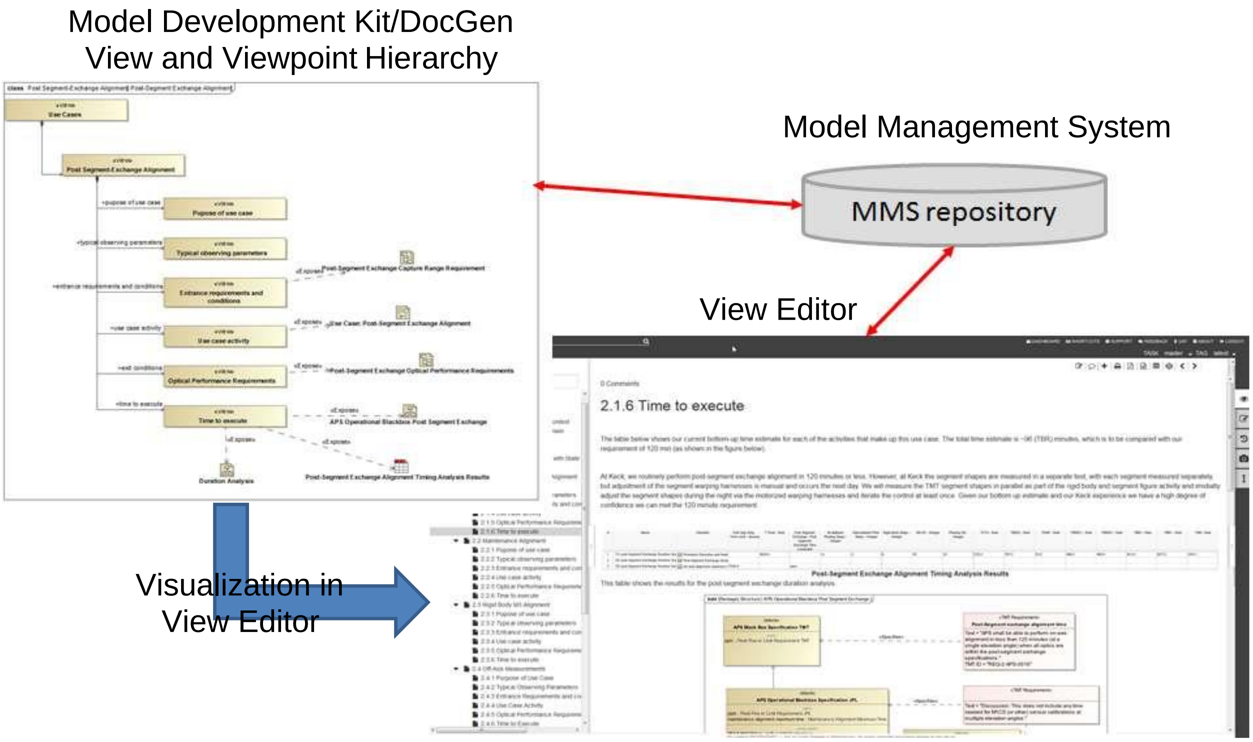


Figure 48. OpenMBEE Core Elements

7.2 MODEL DEVELOPMENT KIT AND DOCGEN

Benjamin Kruse has provided a number of talks and demonstrations covering the following topics:

- Concepts for DocGen as architecturally represented in Figure 49
- View and Viewpoint Hierarchy
- Workflows
- Best Practices and considerations
- Model Findings and System Reasoner supported by MDK
- Usage & Purpose
    - Extracting information for various stakeholders
    - Demonstrated example for AVCE iMBE
    - Demonstrated example for UAV
    - Demonstrated example for NAVAIR Surrogate Pilot
    - Thirty Meter Telescope models has a number of example: https://github.com/Open-MBEE/TMT-SysML-Model/tree/master/Presentations

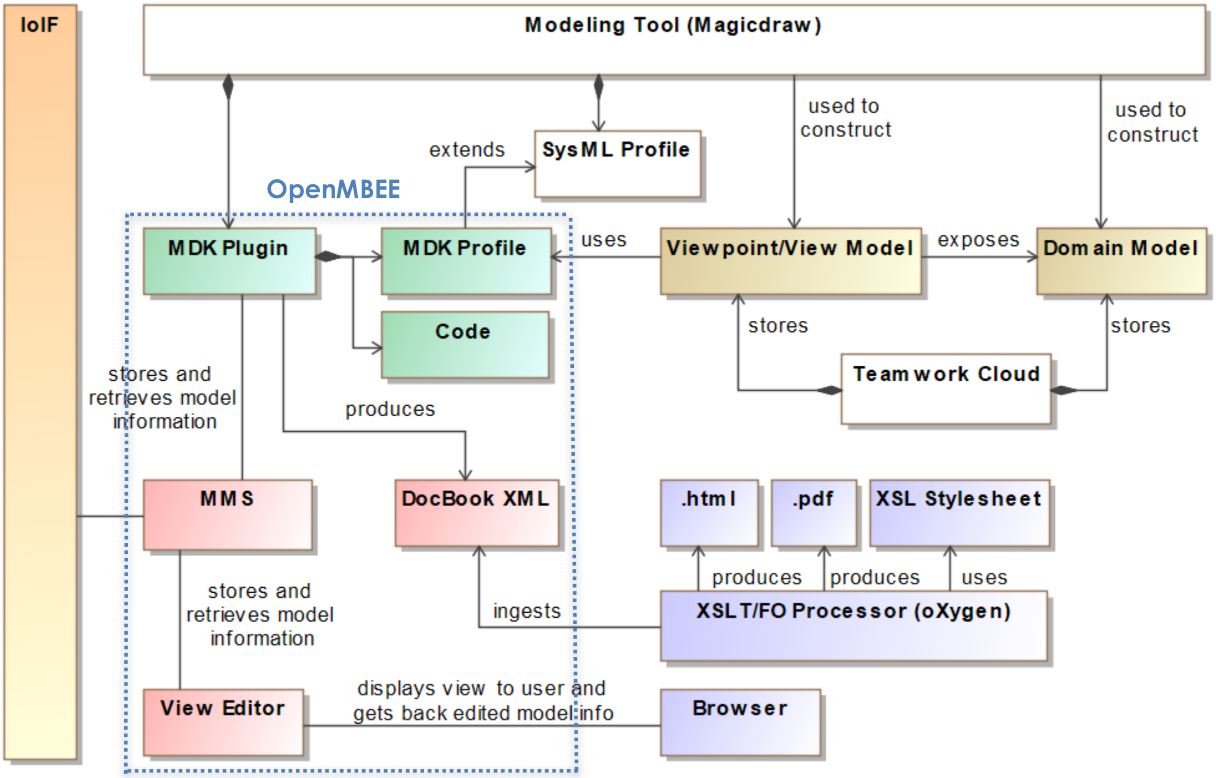


Figure 49. Concepts for DocGen

The basic concepts of a View and Viewpoint hierarchy are shown in Figure 50. There is a profile for DocGen, which includes <<Document>>, <<view>, and <<viewpoint>>. A Document contains one more Views. A View *exposes* Model Content, and *conforms* to a Viewpoint. A Viewpoint is a special type of profiled activity diagram, as shown in Figure 51 that provides a modeling language for extracting information from the exposed view. While this capability was developed to "generate documents" or visualizations from a model, we believe that it can be used for other purposes:

- Use concept to extract parametric values for translating into Monterey-Phoenix 'language' – related to RT-176
- Use concept to extract workflow information to support the Assessment Flow Diagram as discussed in Section 6.7

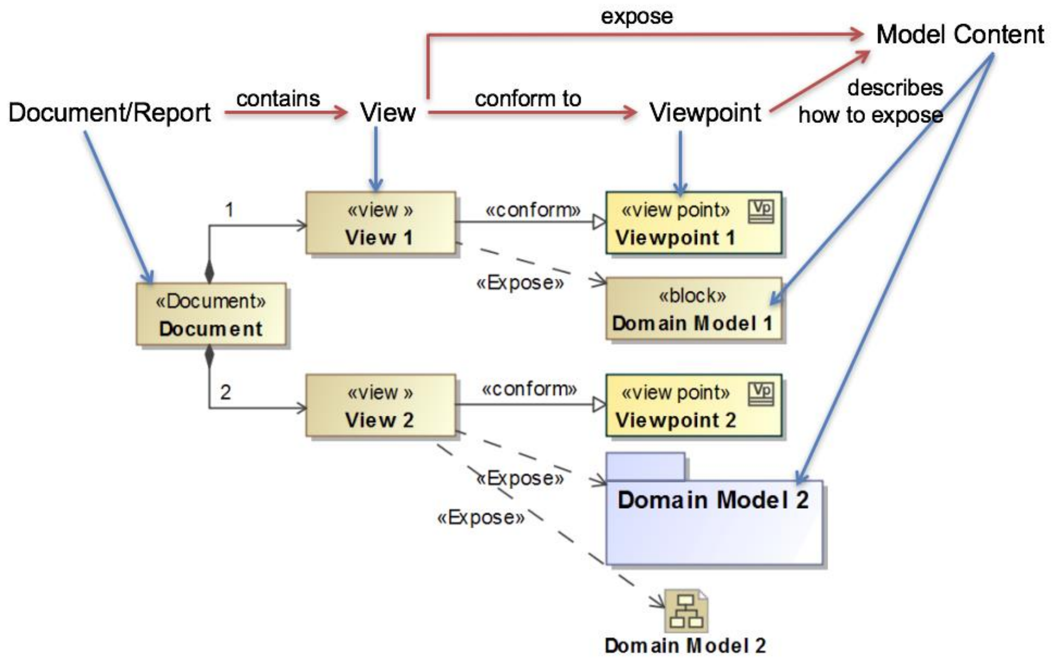


Figure 50. Concepts of View and Viewpoint Hierarchy

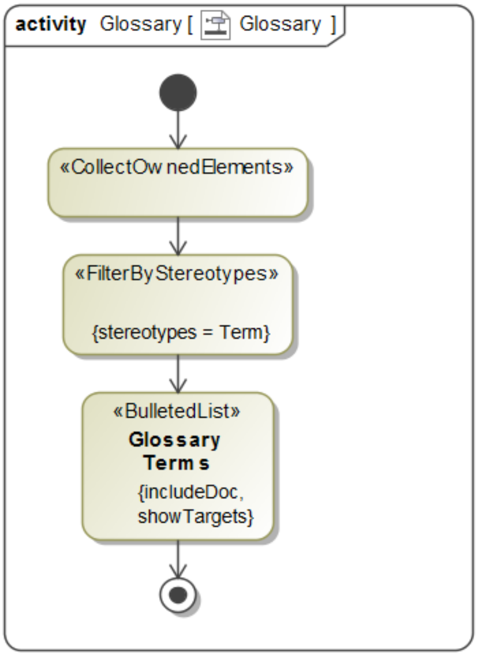


Figure 51. Simple Viewpoint Example

There are a few considerations and best practices for developing view and viewpoint hierarchies for use with DocGen:

- There a number of pre-defined viewpoints, so review those provided in the profile to understand what is available, and to provide guidance in making custom viewpoints
- Expose model elements that align with viewpoints and vice versa
    - Required data must exist in model (e.g. traceability links between elements)

  - Consistent model structure makes data accessible (e.g. nested package structures or existing diagrams at expected position)
- Ordering of sections/views
- Order of sections/views conforms to order of a set of part properties as reflected in Figure 52, which shows partial representation of View and Viewpoint hierarchy for AVCE iMBE (DocGen plugin only displays it through numbered naming)
  - Create sub-chapters through nested views to reduce change impact
- Data representation
  - Produce SysML matrixes only as images or tables
  - There are issues to export simulation plot data
- There is a simulation capability
  - Expected use for web editor (e.g. to recalculate values)
  - Execution of simulation within SysML during report generation, not working as expected
- Viewpoints can be described with the Object Constraint Language (OCL) (as opposed to the activity diagram language)

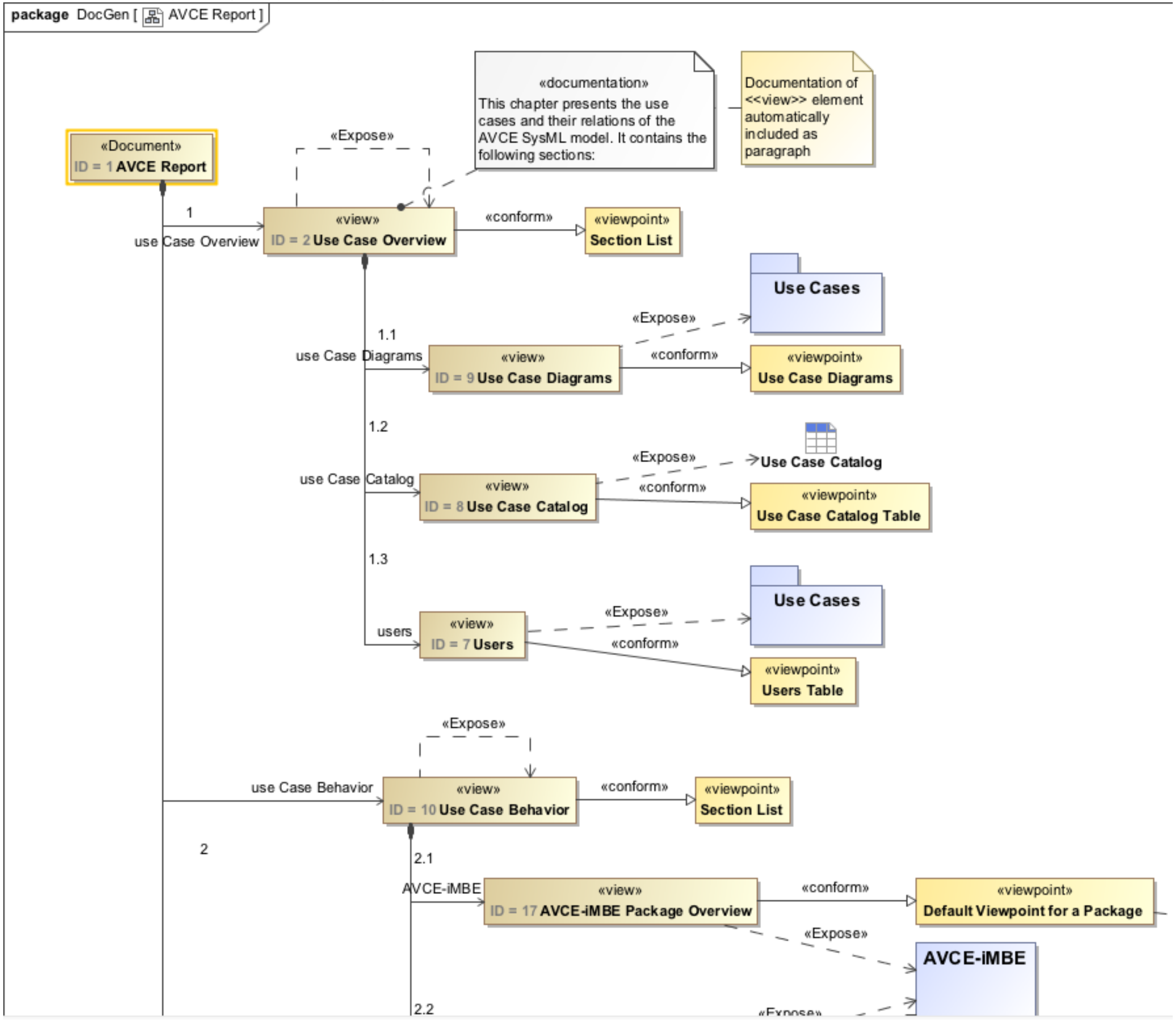


Figure 52. Partial Representation of View and Viewpoint Hierarchy for AVCE iMBE Model

7.3 VIEWS AND VIEWPOINTS

The basic elements, as shown in Figure 53 can be included within an overarching document, which includes:

- Document – the overarching model element
    - Document can include other documents, which also provides another level of modularization and support for reuse
- View (there can be one or more views in a document)
- A View uses the Exposes relationship to associate the View with some element in the model (e.g., Package, Diagram, etc.)
- View conforms to a Viewpoint
- Viewpoint in Model Development Kit (MDK) is a special language created out of a profiled activity diagram that can collect, filter, and then produce a document through a DocBook standard

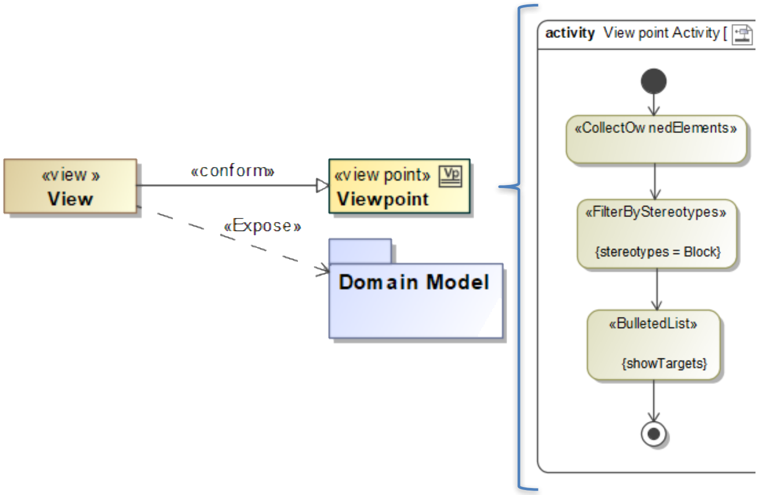


Figure 53. Element of View and Viewpoints

A document assembled from a number of Documents or Views can be generated into DocBook, which can then be generated into PDF, Word, HTML, and other formats. However, these Views can also be "pushed" into the OpenMBEE Model Management System (MMS) as shown in Figure 54. The View Editor can then be used to view the generated specification; in addition, it can export (generate) into Word, PDF, and HTML. The View Editor also allows for editing and updating a generated view that can also be pushed back into the MMS, as well as back into the model (for certain types of model elements).

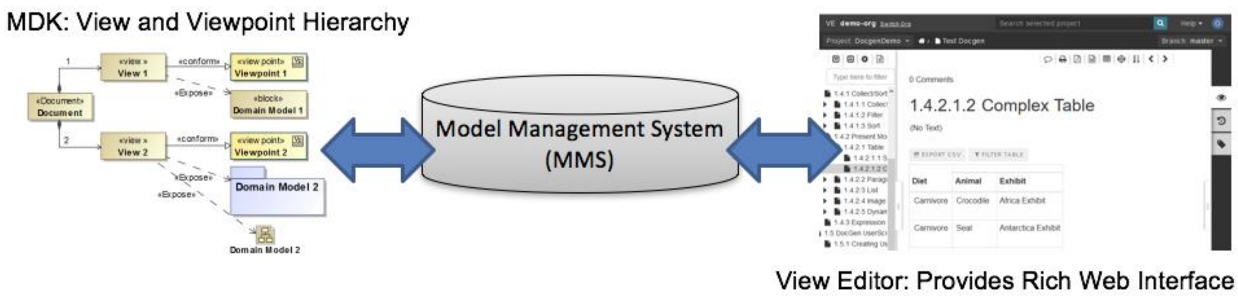
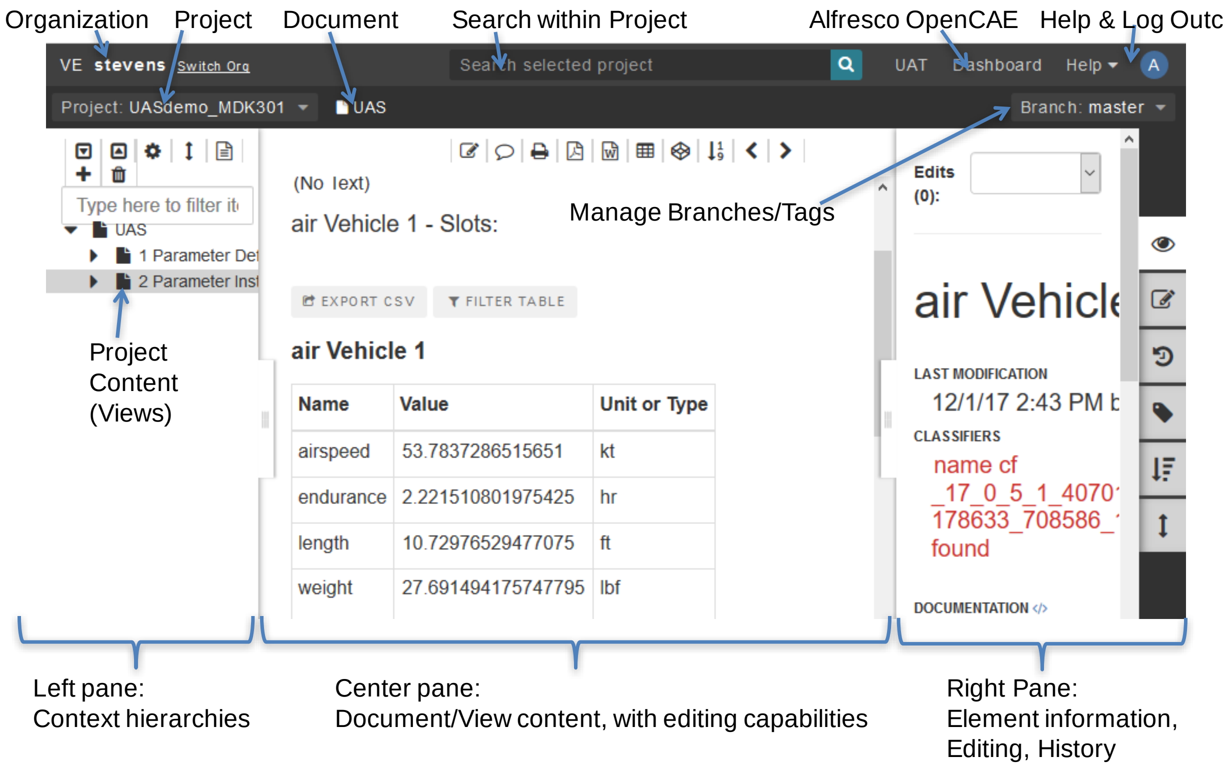


Figure 54. Views are Pushed into Model Management System and Viewable through View Editor

As shown in Figure 55, the View Editor runs in a standard browser and lets users navigate the View hierarchy, and visualize specific Views within the hierarchy, edit the views and examine history associated with changes of the View. There are capabilities for branching those changes. This is part of the future research to investigate the combination of facets related to View and Viewpoint hierarchies, model management in MMS as well as in Teamwork cloud. We are expecting some support from NASA/JPL who is developing some type of guidelines, and working in conjunction with our NAVAIR sponsors on the best methods for model management.



(No Text)

air Vehicle 1 - Slots:

air Vehicle 1

Name	Value	Unit or Type
airspeed	53.7837286515651	kt
endurance	2.221510801975425	hr
length	10.72976529477075	ft
weight	27.691494175747795	lbf

Figure 55. View Editor

7.4 SYSTEM MODEL IN SYSML

We are also using MBSE to model our project, as reflected in the initial use cases shown in Figure 56. We are developing several UAV examples, both for this project as well as for our NAVAIR research. We plan to leverage models between the projects, where possible. For example, as shown in Figure 56, the system domain shows the various elements associated with surveillance, which is shown in a Block Definition Diagram (BDD). We will elaborate on parts of this domain that map back to both mission and system simulation in UC01, UC02, and UC03. They have been made available to our sponsors and our team to use in sandboxing when getting started with OpenMBEE, ViewEditor, MMS, and Teamwork Cloud (TWC).

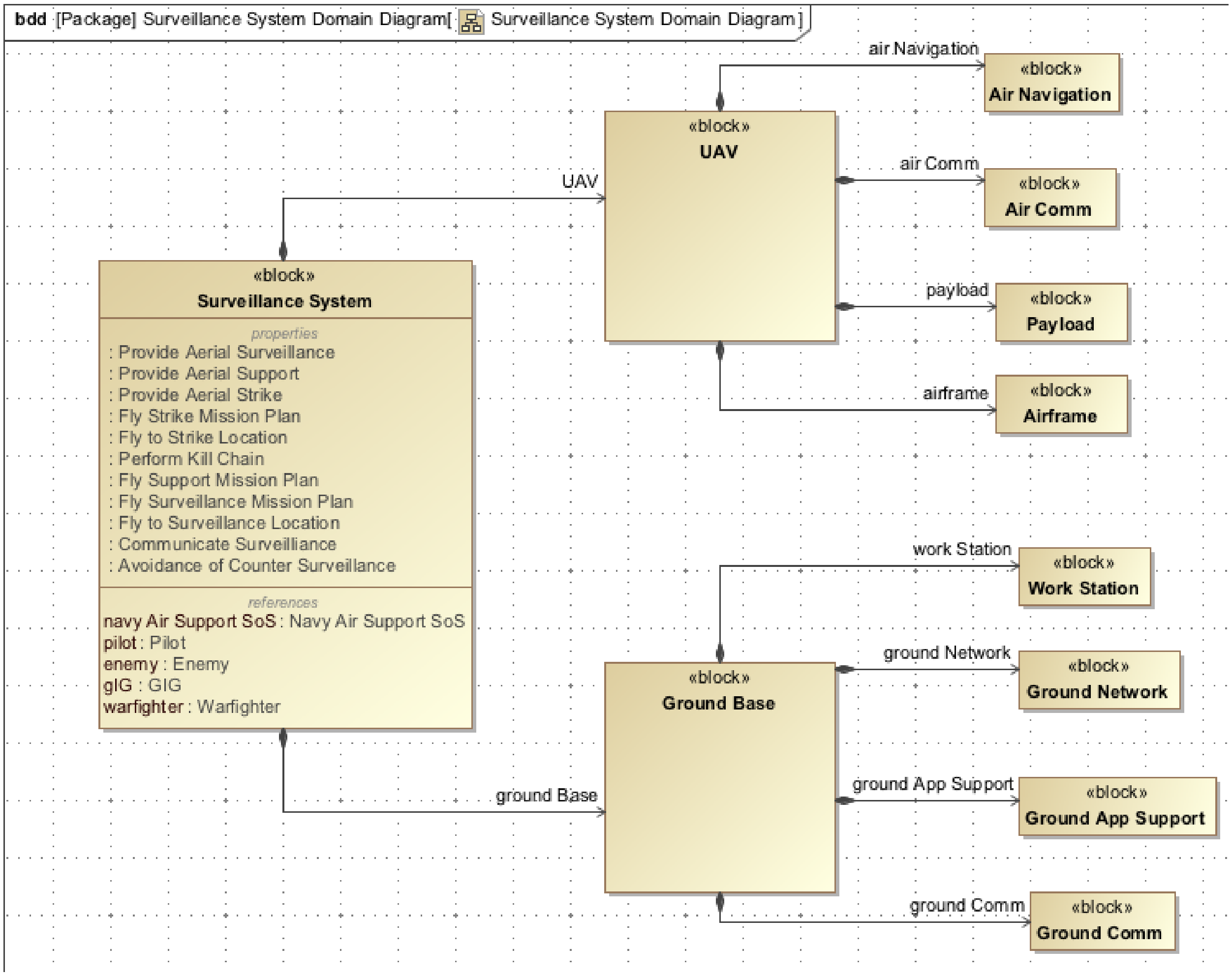


Figure 56. Surveillance System Domain Diagram

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 57 [188]. 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.

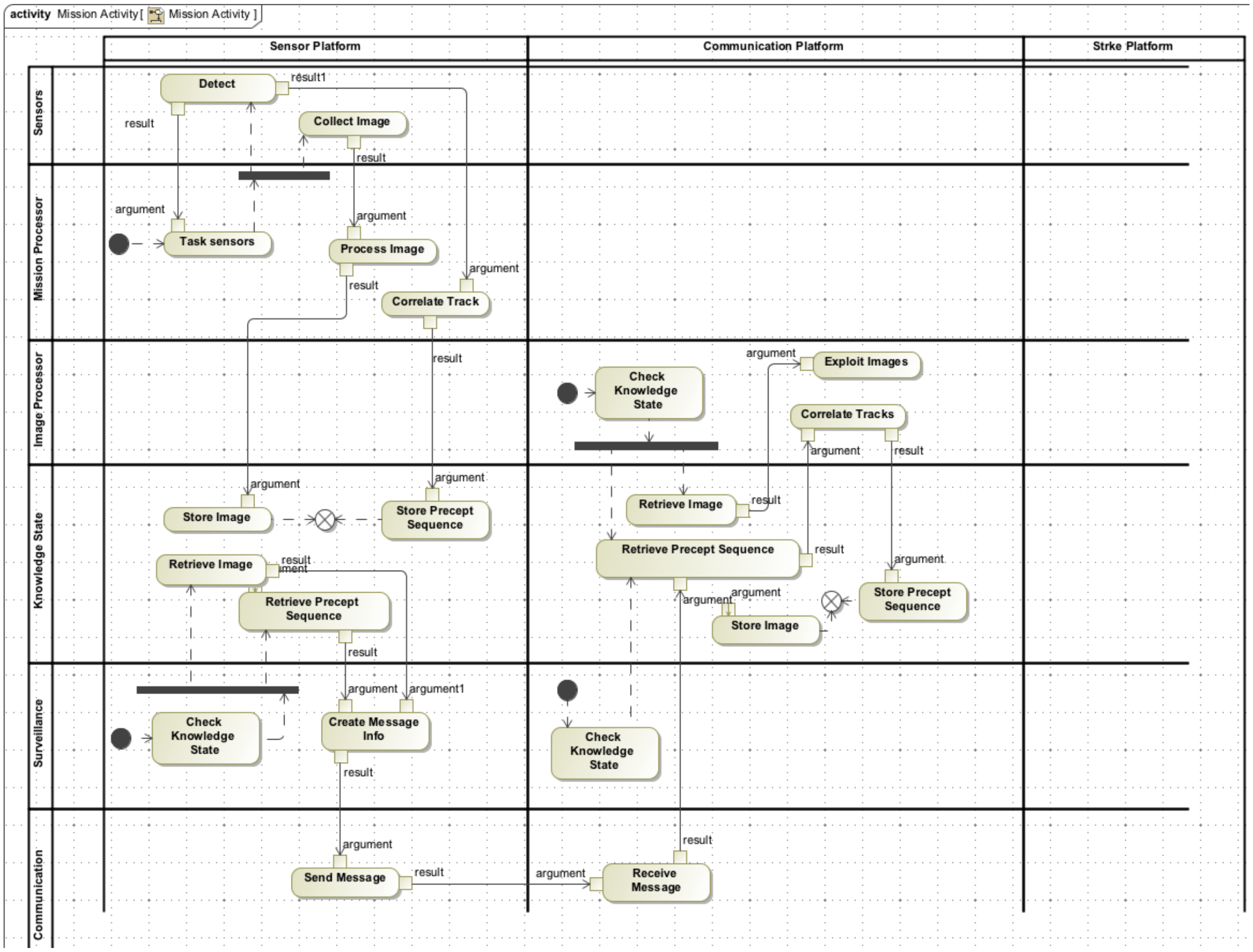


Figure 57. Mission-level Activity Diagram with Swim Lane Partitions

We can further refine the model, and we also have examples that are based on a product family of UAV being developed by our research collaborator, Dr. Russell Peak, under RT-170 that include:

- Rotor UAV 2.1 portfolio effectively completed
  - Includes optical camera option to original package delivery UAV squadron
  - Includes physics calculations via SysML parametrics (par)
  - Includes behavior simulation via SysML state machine (stm) / activity (act) / parametrics (par)
- Fixed-wing UAV 0.1 portfolio initiated
  - Inspired by fixed wing surveillance
  - Applying ~same approach as for rotor UAV portfolio
- We could use Dr. Cilli's UAV example

Some of work in progress elements include the system model for the Fixed-Wing Refueling UAV. These are shown below in a SysML BDD, which shows some of the subsystems of the UAV that include: propulsion, fuel, and refueling subsystems.

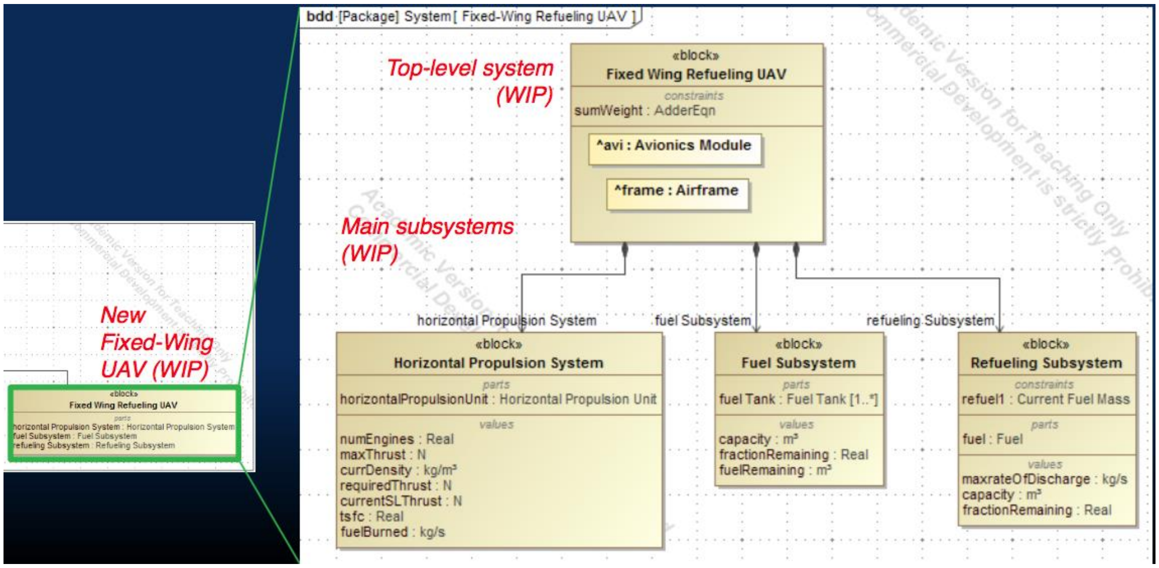


Figure 58. Fixed-Wing Refueling UAV Extension to UAV Portfolio

There are elaborations on some parameters of the fuel system as shown in Figure 59 to do some analysis on the First-Order Physics using SysML Parametrics. A parametrics diagram provides a way to describe constraints between parameters. Add-on analysis tools can then be used to verify that the constraints are satisfiable (i.e., not contradictory). This model is developed in MagicDraw [143], and uses some automation provided by a MagicDraw plugin called the Cameo Simulation Toolkit for requirement verification as shown in Figure 60. For example, the result of pass/fail on a constraint can be traced directly back to specific requirement object in the model.

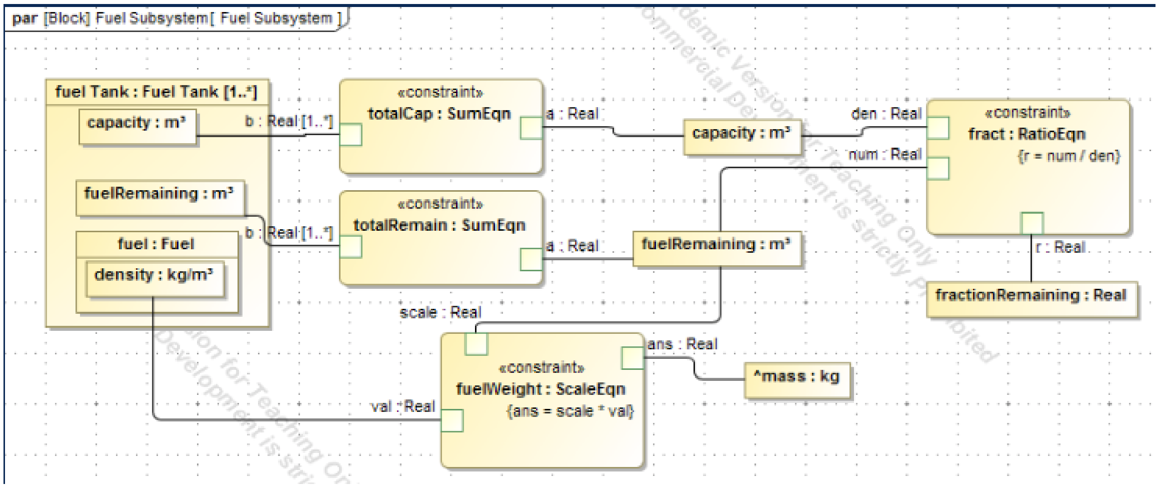
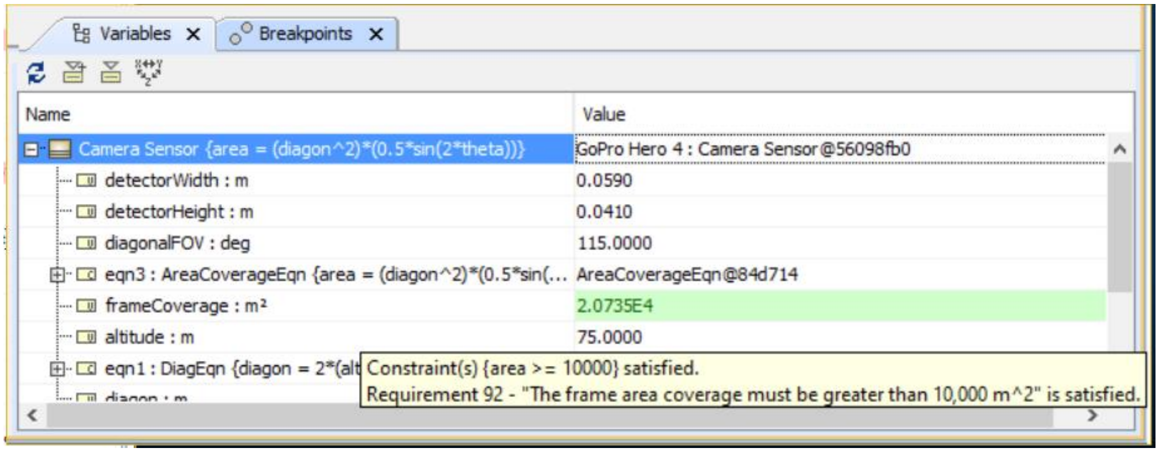


Figure 59. Parametric Diagram of Fuel System



Name	Value
Camera Sensor {area = (diagon^2)*(0.5*sin(2*theta))}	GoPro Hero 4 : Camera Sensor@56098fb0
detectorWidth : m	0.0590
detectorHeight : m	0.0410
diagonalFOV : deg	115.0000
eqn3 : AreaCoverageEqn {area = (diagon^2)*(0.5*sin(…	AreaCoverageEqn@84d714
frameCoverage : m²	2.0735E4
altitude : m	75.0000
eqn1 : DiagEqn {diagon = 2*(alt	
diagon : m	

Constraint(s) {area >= 10000} satisfied.
Requirement 92 - "The frame area coverage must be greater than 10,000 m^2" is satisfied.

Figure 60. Cameo Simulation Toolkit Verifies Constraints Representing Numeric Requirements

We will elaborate on these models to map to UC01, UC02, UC03, and also are investigating the integration of other modeling capabilities such as MathWorks [116] Simulink and MATLAB for UC05.

7.5 NEXT STEPS IN SYSTEM AND MISSION MODELING METHODS

Some of the next steps in modeling methods, which are very much related to using SysML modeling are:

- Methods from traceability from CONOPS, KPP, mission models, to system models, to subsystem models – this may have implications on security, which relies on
- Methods for Model Management using MMS and TWC
- Methods for modularizing models, which is directly related to model management and impacts traceability
    - Early demonstrations in the NAVAIR Surrogate pilot have used the Project Usage capability to include a mission model into a system model, where the traceability from the system model to the mission requirements is traced
- Methods and best approaches for using access rights and authentication
- Methods for multi-level security; this is in scope, but not funded

8 COUNTER UAS IN THE CONTEXT OF MODEL BASED ENGINEERING (UC05)

This use case develops both the Model-Based Engineering (MBE) methods, the counter Unmanned Aircraft Systems (UAS) scenarios and evolving approaches to Automated Concurrent Engineering, specifically related to MBE and manufacturability. This use case may be split in the future. In the context of working with the physical representation of various elements that are characterize abstractly in the system model, in the mechanical and electrical space, we are infested in how MBE can improve the physical reliability through manufacturing. Therefore, Kishore Pochiraju has discussed:

- Representation Methods, Model Frameworks and Verification Tools for Cyber Physical Design, which are discussed more in Sections 8.2 and 8.3
- Automated Concurrent Design as discussed in Section 8.4

8.1 MODEL-BASED ENGINEERING

We distinguish MBE from Model-Based System Engineering (MBSE). Typically, MBE involves modeling and simulation capabilities related to specific disciplines, electrical, mechanical, software, and the potential use of domain-specific modeling tools. Most importantly, we are interested in how these modeling tools for a specific, some of which have analysis and simulation capabilities, can be integrated with mission and system-level modeling and simulation (e.g., UC01 and UC02), MBSE (UC04), and MDAO (UC03). These various type of modeling capabilities are fundamentally important for a new class of systems that are generally referred to as Cyber Physical System (CPS).

8.2 MBE AND CYBER PHYSICAL SYSTEMS (CPS)

The phrase "cyber-physical systems," coined by Helen Gills [84] defines "physical, biological, and engineered systems whose operations are integrated, monitored, and/or controlled by a computational core. Components are networked at every scale. Computing is embedded into every physical component, possibly even into materials. The computational core is an embedded system, usually demands real-time response, and is most often distributed." Based on a 2015 National Academy of Science preliminary report [136] "Cyber-physical systems (CPS) are increasingly relied on to provide functionality and value to products, systems, and infrastructure in sectors including transportation (aviation, automotive, rail, and marine), health care, manufacturing, and electrical power generation and distribution. CPS are smart, networked systems with embedded sensors, computer processors, and actuators that sense and interact with the physical world (including people), support real-time, guaranteed performance and are often found in critical applications. Clearly, the types of UAS that are of interest to ARDEC are CPS.

Kishore Pochiraju presented his research entitled: *Representation Methods, Model Frameworks and Verification Tools for CPS Design* in a bi-weekly session. Some of the challenges discussed involve uncertain computation and network delays/latencies that can disrupt control performance and plant stability. Such control performance is critical to maintain system compositionality across these vary disciplines of a CPS. The integration of MBE tools with MBSE tools is of particular interest.

Another important aspect is CPS applications involve components that interact through a complex physical environment. Reliability, security, trustworthiness poses particular challenges in this context. These CPS need to be highly dependable, reconfigurable, and in many applications, certifiable. Trustworthiness must also extend to the system level.

8.3 COUNTER UAS

We have included the counter UAS use case in this section, because Kishore has other related research in his area of expertise. To summarize the key objective for this counter UAS problem:

- Given a counter UAS system that identifies and restricts the flight of a UAV in a specified space
    - Represent the system using a compositional framework and appropriate models, much of which has been summarized in Section 8.2
    - Validate
        - For example, analyze the abstraction for a requirements satisfaction

        - Predict
            - Performance degradation due to timings and time-delays in implementation
            - Analyze the composability of the components and dependence on the system performance
            - Analyze the compositionality of the entire system
    - Basic premise:
        - Watches the space for a UAS (Sensor Component)
        - Locates the UAS dynamically (Localization component)
        - Defeats the UAS (Act component)

To put the magnitude of this challenge is perspective, Business Insider magazine [45] reports the global aerial drone market will reach nearly $13 Billion USD within the next 10 years. Nearly 75 percent of this market is projected to be defense-related. The commercial marketspace is also expected to develop into a $3 Billion market in the very near future. UAS will surely be ubiquitous enough to be perceived as annoyances or worse, threats, in many spaces [62]. The counter UAS technologies address the need for defense against unwanted UAS in military and public spaces and for the enforcement of various regulations against drone flight.

The current counter unmanned air systems, depending upon the context, integrate various "*Watch, Match and Catch*" methodologies [66]. These systems include area surveillance to detect the presence and location of a signal (watch phase), match (associate) to a UAV signature, and deploy a catch or defeat technique such as jamming. The watch phase can be based on passive detection of electromagnetic, thermal (IR), acoustic signatures or active use of RADAR [130], LIDAR, acoustic beamforming [164], and optical tracking methods. Match phase entails the use of library of signatures, machine learning methods, and physics-based models to identify the presence of a UAV in the surveilled space and also detect its type. Catch or defeat [17] requires jamming control signals or physically affecting the flight of the UAV with nets, projectiles or other UAS. Use of a particular method for the catch phase may be rendered infeasible due to safety requirements and the risk for collateral damage.

Modeling is central to all three phases. Watch phase technologies employ modeling not only for enhancing the signal to noise ratios, extracting localization information and constructing 3D representations of the tracked target, but also for numerous other purposes. Matching is typically conducted based on pattern identification models with the support of datasets. Catch methods employ modeling for interception path planning and directing transmission antennae for directional and selective propagation of jamming signals.

Due to the real-time nature of all the three problems, most accurate models that have the necessary computational efficiency are generally preferred. Accuracy versus time for the computation of a model-based solution is the general trade-off while deciding on the best algorithm to implement in each phase. The three phases are typically distributed among heterogeneous subsystems with some pipelining of the tasks. However, the total time for response (detect-to-defeat) will be the sum of watch, match and catch phase times.

The objective of this sub-task is to investigate the role of modeling in both in terms of the effectiveness (i.e. accurately watching, matching and catching) and the performance (i.e. total response time and availability times).

Three activities are proposed for this use case:

1. Analysis of models used in counter-UAS methodologies: This activity entails analyzing open-military grade or a commercial counter UAS system and mapping role and performance of the models in the system. The expected deliverable is a broad state-of-the-art and gaps analysis report.
2. Assessment of watch-match phase models: The sub-task team will consider LIDAR and acoustic-based detection technologies and the effectiveness and performance of the models in the watch and match phases of the counter UAS problem.
3. Limited field experimentation: This sub-task team will collaborate in the design and conduct of preliminary experimentation of an idealized counter UAS system. The objectives for the experiments will be:
    - Measure the performance of selected signal detection and UAV localization models used in watch/match phases.
    - Investigate models that enable selective defeat (jamming) of one UAV flying in homogeneous or heterogeneous swarms.

With the intent of isolating and measuring the role and impact of the models, the experimentation will be planned in uncluttered physical spaces and with known dynamical behaviors of the UAVs.

8.4 AUTOMATED CONCURRENT DESIGN

Kishore provide two talks extending the first talk on CPS to reflect back on how the formalism and semantically rich information can contribute to automated concurrent design. The two talks included:

- Knowledge-Based Product Design and Manufacturing in the context of Automated Concurrent Engineering System (ACES) Technologies to provide significant reduction in product development time and cost while optimizing the design and its manufacturing.
    - This was prior research, but there is a type of metaphor, where this work in the more mechanical space represented design knowledge to ensure manufacturability; we are attempting to do somethings similar in the system, system of system, and mission space.
- Design Automation also related to Automated Concurrent Engineering Approaches
    - This particular research extended the prior work by investigating the feasibility of formalizing the design process to “provide a robot with a set of ‘specification’ to provide a design automatically.”
    - This specifically formalizes a system as a network of dependencies from requirement to design controls.
    - Provided early approach to MDAO for tradespace exploration.
    - Networks of formalized design information allow design automation to proceed through a search process that can now be enhanced by Machine Learning and Deep Learning techniques and algorithm.

8.5 ARCHITECTURE AND PROTOTYPING OF SYSTEM SIMULATION WITH SEMANTIC DATA EXCHANGE

The concept of a network of design dependencies can be characterize in SWT. The RDF which represent specific model instances, and are aligned with an ontology, is a graph (network of dependences). The concept is to create "gate keeper" tools that create/manage semantics and provide semantic data services to simulation tools. These gate keeper tools (two of them to be prototyped) differentiate data store/retrieve tasks into concurrent add/edit/modify operations on knowledge and data stores. The operations are divided into knowledge-dependent (may require negotiations with Human/AI experts), policy dependent (require reasoners - from heuristics, policy statements), and simply tedious tasks (i.e., automated out - e.g. use of a dictionary/thesaurus to check typos). The tools then create appropriate workflows. We also use the concept of "regularized operations" meaning all operations on knowledge/data stores complete if the integrities of the stores are maintained.

Kishore is aligning some of his research for semantic data exchange with our IoIF, with the objectives to:

- Create a “simulation-as-a-Service” framework with multi-physics, concurrent and concurrent execution of simulations during system architecture and design process.
- “On demand” and “As Appropriate” trade simulations during various phases of large complex systems design/integration
- Enable service-discovery, data-curation and tool interoperability
- Generalized abstraction for spatial, temporal and stochastic fields *with mapped semantics*
- Framework requirements:
    - Generalized abstraction for embedding simulation tools
    - Simulation concurrency and pipelining
    - Data interoperability
    - Model abstractions enabling substitution
    - Indexed simulation inputs, outputs, storage
    - Abstraction to capture model use in design
    - Dynamic data flow tracking
    - Data model capable of large (2GB) data segments, access control, storage and transport.
    - Agnostic to OS and computational hardware
    - Open Application Programming Interface (API)
    - Support for real-time systems – Real-Time Controller API

8.6 MBE ANALYSIS FOR UAS ENERGY ANALYSIS

In order to accurately evaluate system performance as well as design choice consequences, two areas of battery system modeling have been explored by Andrew Dawson in support of added realistic performance in the quadcopter UAS elements in the graphical CONOPS (UC01). The two analyses include:

8.6.1 MASS TO ENERGY CAPACITY:

Based on the general architecture of common battery systems, system mass was anticipated to vary linearly with energy capacity. Battery parameters were compiled for the catalog of systems available from Gensace and Tattu, which are widely utilized in small-scale UAVs. For these batteries, the expected relationship was confirmed and can be expressed as follows:

$$\text{mass [g]} = 5.472\ (\text{capacity [Ah]} * \text{voltage [V]}) + 61.87$$

The linear regression performed had an R-squared value of 0.991. This relationship allows battery mass to be easily incorporated into a variety of performance and flight models.

8.6.2 VOLTAGE VARIABILITY DURING DISCHARGE:

This analysis examined the relationship between discharge levels and maximum available power. Common battery systems specify a C-value, which is the maximum current that the system can safely produce. It is typically expressed as the ratio of maximum current to the current produced when discharging over one hour (the Ah rating). Therefore, a 10Ah battery with a C-value of 10 could produce a maximum of 100A.

During discharge, battery systems experience reducing voltage as charge level decreases. Therefore, for a specified C-value, the maximum power that the battery can produce will decrease along the discharge cycle. This is critical for UAV performance, because certain flight or performance characteristics may degrade over the mission cycle.

An empirical relationship between normalized discharge level (% of capacity) and voltage level (% of rated) was determined based on typical discharge curve literature. This is only intended to demonstrate the relationship and is not fully representative of all battery systems. Two variants were considered:

1. Increasing C-values impact the amount of voltage sag during discharge
2. Increasing C-values impact both the voltage sag and overall discharge capacity

The equations developed to describe these relationships can be utilized in performance models to determine the maximum available power at any point in the discharge cycle.

9 DECISION FRAMEWORK (UC06)

ARDEC uses the Integrated Systems Engineering Decision Management (ISEDM) Process to improve defense acquisition decision-making, which we generally refer to as the Decision Framework in this report. The ISEDM process addresses the pressing issues targeted by the Department of Defense’s Efficiency and Better Buying Power Initiative and the 7-January-2015 DoDI 5000.02. A central issue confronted by both the initiative and the instruction was that systems engineering trade-offs made between capability requirements and lifecycle costs early in the acquisition process were rarely conducted and consequently realistic program baselines were not established such that associated lifecycle costs of a contemplated system are affordable within future budgets. Through the use the ISEDM Process and the family of synthesized data visualization techniques, systems engineers are able to assess a large number of system alternatives across a robust set of competing objectives in the presence of uncertainty and quickly recognize important trends across cost, schedule, and performance dimensions. While the ISEDM process has been applied with success to a number of defense research and development projects, there are several opportunities for enhancement and extension.

There are several objectives within this use case and several notable accomplishments as discussed earlier in the context of Figure 6. We explored potential enhancements and extensions to the ISEDM process and the related decision support tool, starting with AAMODAT. However, due to potential restrictions on sharing or releasing AAMODAT, we have developed and formalized aspects of the ISEDM, which have a prototype implementation integrated within IoIF.

We leveraged and demonstrate tool-to-tool integrations for the ISEDM process using an example the UAV SysML model documented in papers by Dr. Matt Cilli [49][50]. The first extension formalized the process using SysML, which integrates with ModelCenter, through MBSEPak, to illustrate the MDAO concept (UC03) for alternative analysis [23]. Another extension demonstrated useing OpenMBEE MDK plugin to transfer the UAV SysML information to MMS. These are part of the IoIF capabilities to transform the SysML information stored in MMS into the IoIF SWT to align with a decision ontology. We have created several instantiations of OpenMBEE [150] with a Docker [72] configuration, which is an underlying element of IoIF including MMS and View Editor. The transformed information from MMS, now stored in IoIF SWT is transformed using SPARQL queries into a representation to support visualizations of the various tradeoffs in Tableau [187]. IoIF now provides a substantial foundation for follow-on research and other synergies that have been discussed with our sponsor about elevating the Decision Framework concept in the context of IoIF to mission scenarios, or combinations of mission scenarios given system capabilities that can be composed into mission capabilities. We believe this capability to be applicable to ARDEC, but generally applicable to acquisition organizations such as NAVAIR. These capabilities have been provided in the form of models, Docker scripts, configuration information (e.g., tools/software versions), and software in order for ARDEC to replicate this capability.

9.1 DECISION FRAMEWORK OBJECTIVES

This section reviews some of the objectives that are satisfied through the accomplishments, but also represent extensions to the current capabilities for future research that can now be incorporated to the Decision Framework capabilities in the context of IoIF:

- Generate a library of fundamental objectives hierarchies: A fundamental objectives hierarchy (and its associated measures) describes the criteria by which the goodness of each alternative is assessed. Studies show that the formulation of an objectives hierarchy is a difficult task and is often done incorrectly – significantly impacting decision quality in a negative way. The purpose of this sub-objective is to generate a library of thoughtfully prepared and well vetted objectives hierarchies for a set of common weapon system types such that a systems engineer can use a hierarchy from the library as a starting point that can be easily tailored for the particular decision at hand.
- Develop a Decision Risk Tracker: Cilli [49] identified 40 potential pitfalls associated with systems engineering trade-off analyses and through the use of practitioner surveys measured the perceived likelihood of encountering each pitfall and the consequence to decision quality given a particular pitfall was indeed encountered. The purpose of this sub-task is to develop a methodology to instantaneously assess the overall risk of a systems engineering trade-off analysis project and to update the risk assessment as known pitfalls are avoided through the use of best-practices through the execution of the study.
- Incorporate a Decision Adviser Feature: Create a context sensitive pop-up decision advisor to alert users of best practices associated with the current process step.
- Add context sensitive best practices pop-up wizard to (avoid common pitfalls)

- Create Objectives Hierarchy Library
- Enable Assessment Flow Diagram (AFD) Auto-generation (done)
- Integrate Data Visualization COTS capabilities (i.e. Tableau)
- Integrate Value Scheme Elicitation Tools (part of the SysML method to formalize AFD
- Develop improved automated Swing Weight Matrix Generator
- Integrate Conjoint Analysis tool
- Integrate DOE capability to generate run matrix for agent-based models
- Enable automated Design Structure Matrix (DSM) generation and link to IRL portion of schedule estimator module
- Use unclassified, public releasable, but plausible and data rich problem (sUAV case study developed under ERS effort) to demonstrate ISEDM best practices with new upgrades listed above (done as part of UAV example)
- Solve same problem but purposely trip on identified pitfall to illustrate why ISEDM process that avoids pitfall is superior

The current status and new plans include, but are not limited to during the completion of Phase II:

- Formalization of Assessment Flow Diagram using SysML, MBSEPak, and ModelCenter demonstrated (done)
- Decision ontology maturing to align with Decision Framework (in process)
- Release of the measure and metrics computation embedded IoIF/Decision Framework to be published
- Tools for populating Triple Store that is ontology compliant (in process)
- Plan to create SPARQL queries from Triple Store that incorporate the measure and metrics computation to produce input to visualization tool (e.g., Tableau) (version 1 demonstrated at working session #10)

9.1.1 DECISION FRAMEWORK METHODS

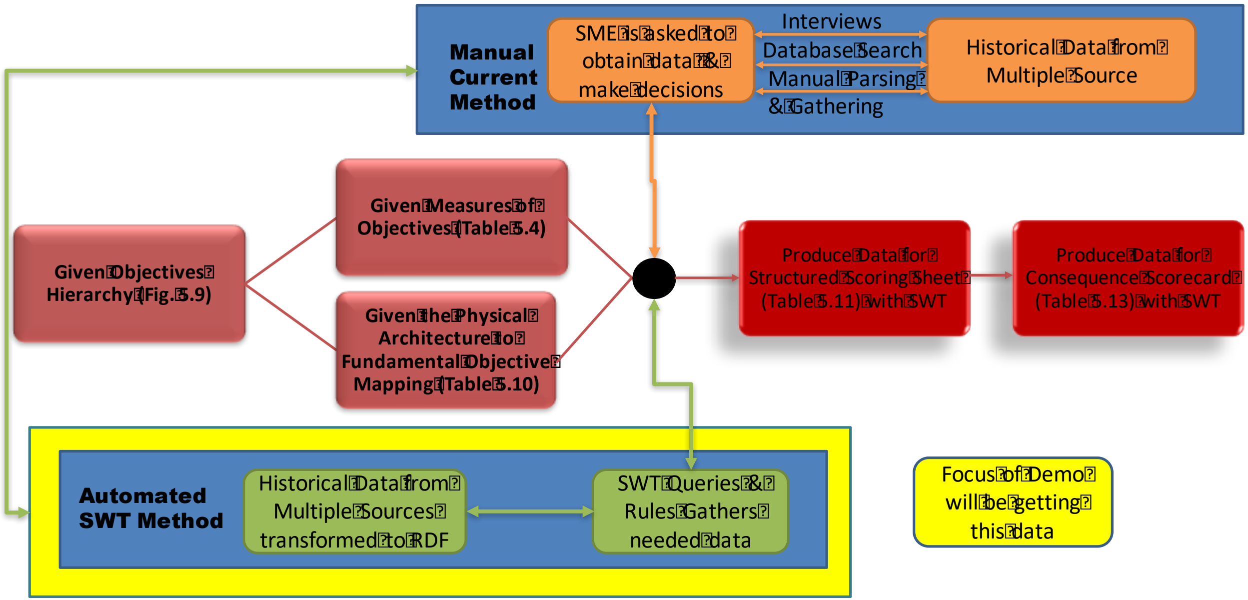
Research methods to achieve stated objectives have used product development case studies approved for public release, which is represented in a book chapter created by Matt Cilli [50]. In response to a request from the Engineered Resilient Systems (ERS) program, ARDEC is generating a hypothetical yet plausible case study that can be used to stimulate and focus academic discussion regarding systems engineering tradeoff analyses in the context of new product development efforts. The case study possesses elements of story such as setting, characters, plot, conflict, and point of view (Omniscient Limited), and theme. It will also provide detailed narrative incorporating many viewpoints; involve ambiguity, uncertainty, and un-structured presentation of initial information; give rich description of potentially useful data at multiple levels of fidelity; allow for multiple outcomes; and be publically releasable.

9.2 USING SEMANTIC WEB TECHNOLOGIES TO FORMALIZE DECISION FRAMEWORK

We have formalized the ISEDM process as part of IoIF as shown in Figure 6, and are working to create other related use cases as reflected in Figure 3, but extending the decision ontology with other interoperability ontologies. The method embodied in the SysML UAV example formalized the Decision Framework using an ontology with computational support provided by SWT and visualization tools in Tableau that support related perspectives on the system (UC01, UC02, UC03, UC04, UC05). These elements would include:

- Objective hierarchies
- Value functions
- Assessment Flow Diagrams (AFDs) trace the relationships between physical means, intermediate measures, and fundamental objectives, as discussed in Section 6.7
- Uncertainties

Figure 61 shows the use case refinement that has been discussed by the team. Mary Bone has provided an extended session at the third working session on a concept to show how SWT could support this effort to populate AAMODAT or other computational mechanism using SWT that can be part of IoIF.



"Wiley: Trade-off Analytics: Creating and Exploring the System Tradespace - Gregory S. Parnell." [Online]. Available: http://www.wiley.com/WileyCDA/WileyTitle/productCd-111923753X.html. [Accessed: 27-Mar-2017].

Figure 61. Decision Framework Use Case Refinement

There have been other developments toward that:

- Robin Dillon-Merrill is working on templates for different type of objective hierarchies (e.g., portfolio, product)
- Matt Cilli has an update to the UAV case study [50]
- Mary Bone walked through the use case using example to show how to use SWT (UC10) to produce score sheet and consequence score card for objective: reach areas of interest quickly
  - For demo purposes, Mary used SWT to get example data from DBpedia (which is a crowd-source effort to extract structured information from Wikipedia and make this information available on the Web)

    - Created a simple Aircraft Ontology & Properties for demo to show semantically rich data extracted from DBpedia using SWT tools (Protégé, OWL Viz, RDF)
    - More details in UC10.
- John Dzielski has formalized the Assessment Flow Diagram concept using SysML, MBSEPak, and ModelCenter as shown in 6.7.
- Tom Hagedorn and Ian Grosse have developed a Decision ontology underlying Decision Framework.

As discussed in Section 6.7, we also noted that the AFD is probably the single view that best describes how the specific design choices are made across the product structure, and are transformed into consequences across the fundamental objectives through an array of interrelated models. Because of the similarity in the AFD to MDAO workflows. We are researching ways to model the AFD as an MDAO workflow, because those workflows would most likely be related to Key Performance Parameter (KPP). We had noted in the past that the Decision Framework would potentially support a method for deciding on the KPPs. The AFD might prioritize the needed workflows to defined using MDAO (e.g., ModelCenter).

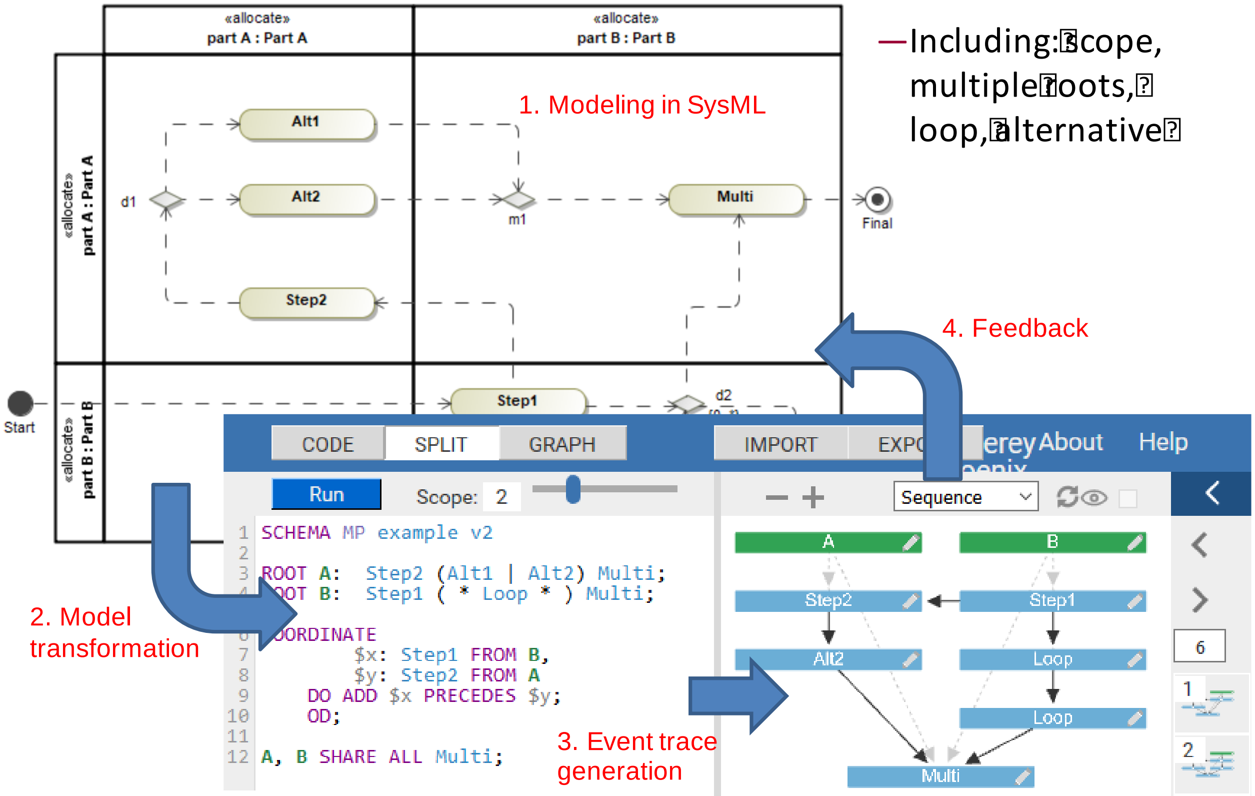
10 MCE IMPACTS ON VERIFICATION AND VALIDATION (UC07)

There was no explicit task to support Verification and Validation (V&V), however MCE can inherently produce information in a more formal way that can enable early and continuous V&V. Rigorously defined models can directly support V&V, and this could both subsume cost and risks. This use case can likely identify candidate requirements for AVCE. Therefore, we added this use case as a place holder, and are considering a potential task that relates to both UC05 and UC03. There are a number of possible contribution to various types of V&V. For example, the effort to use SERC RT-176 effort of Monterey Phoenix for V&V of requirements may support some of this effort. The model created by Georgia Tech for RT-170 has other examples illustrating some V&V. If we are able to use the IMCE ontologies for systems engineering from NASA/JPL, then this would provide another avenue to support V&V.

10.1 REPRESENTATION TO FORMALIZE MONTEREY PHOENIX FOR REQUIREMENT VERIFICATION AND VALIDATION

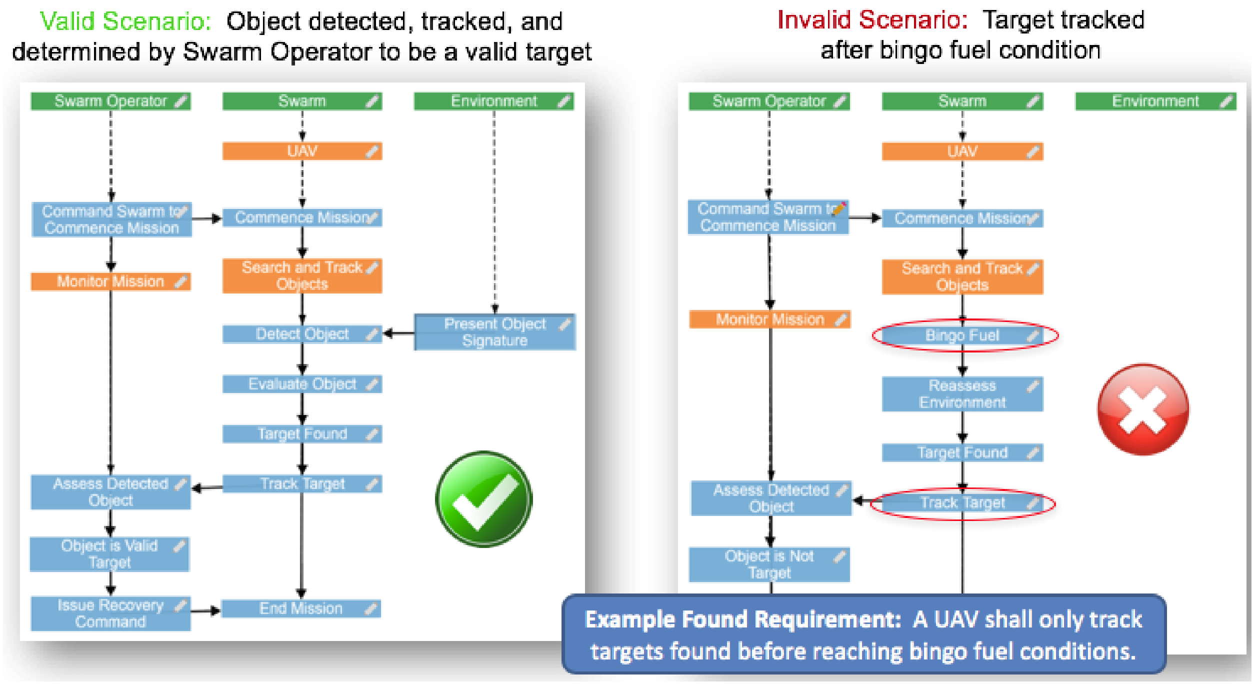
We have started investigating the development of SysML representations to formalize the Monterey Phoenix (MP) research under RT-176 to support requirement verification and validation [81] [82]. MCE does provide some unique opportunity to be more effective at contributing V&V evidence in early design. Rigorously defined models can directly support V&V, and this could both subsume cost and risks.

The results accomplished against this effort to use SERC RT-176 effort of Monterey Phoenix for V&V of requirements is showing progress. The basic concept is to formalize using SysML graphics, and in this case activity diagrams and then transform into the MP language [129] as shown in Figure 62. MP then uses the formal language to generate graphical representations of the behaviors, as shown in Figure 63 [161] that can be derived from the language of the formalized behavior to a given scope level (e.g., Scope 2 in Figure 62). The verification step does require a person to check the different visual behavioral representations for correctness. This concept is similar to model checking.



```
SCHEMA MP example v2

ROOT A:  Step2 (Alt1 | Alt2) Multi;
 OOT B:  Step1 ( * Loop * ) Multi;

 OORDINATE
        $x: Step1 FROM B,
        $y: Step2 FROM A
    DO ADD $x PRECEDES $y;
    OD;

A, B SHARE ALL Multi;
```

Figure 62. Representation and Transformation from SysML Activity Diagrams to MP



Example Found Requirement: A UAV shall only track targets found before reaching bingo fuel conditions.

Figure 63. Generated Visualization of Scenarios by Monterey Phoenix

11 ACCESS AS CHIEF ENGINEERING ROLE (UC08)

This use case was created so that one of our researchers, experienced in systems engineering can provide some level of assessment of our overarching approach and contribute to the requirements for AVCE. We too want to bring as many technologies as possible into our lab at Stevens in order to assess the gaps, but are also interested in bring in master's students to use methods derived from this research. Our first experience has been using the OpenMBEE and the Model Development Kit (MDK)/DocGen in our third course of a four-course series [193] on Cyber Physical Systems. The Stevens SYS-673 course on *Implementing Cyber Physical Systems: Bring Solutions to Life* [24] demonstrated the use of MDK/DocGen but two teams to fully generate their final technical reports directly from their SysML models. There examples, in addition to demonstrating system modeling, also showed the use of hazard, fault tree, and FMEA analysis using the Cameo Safety and Reliability Analyzer Plugin. The generated models are available for review through the OpenMBEE View Editor on our Amazon Web Service website.

12 TRADEOFF ANALYSIS OF TECHNOLOGIES FOR INTEROPERABILITY OR INTEGRATION (UC09)

This use case as discussed throughout this report has emerged as an organizing element for many of the research use cases. Originally the research was focused on supporting requirements analysis for AVCE iMBE and to demonstrate new concepts for using interoperability through SWT to achieve tool-to-tool integration. Specifically, we have looked at, and continue to look for alternative technologies and tools used by ARDEC, as well as other organizations who are creating and evolving their integrated modeling environments. We have a laboratory to support research on the tradeoff analysis of technologies for integration or interoperability in order to further study the technologies and provide demonstrations. Most importantly, the IoIF framework is evolving, and we have provided several demonstrations for both integration and interoperability through SWT (UC00, UC01, UC02, UC03, UC04 and UC06) as shown in Figure 6.

12.1 INSTANTIATION OF AUTHORITATIVE SOURCE OF TRUTH

We created a Docker [72], as reflected in Figure 64 that allows us to rapidly deploy OpenMBEE. Docker [72] has a number of capabilities, but we were able to develop a "Docker script" [73] that is able to pull together and configure all of the technologies (e.g., Alfresco content management, Tomcat database, Elastic search, etc.) needed to deploy OpenMBEE. This allows to us to work on the underlying capabilities for demonstrating IoIF in a distributed Authoritative Source of Truth (AST). We continue to focus on SWT, including, relationships between OpenMBEE, Model Management System (MMS), View Editor, in the instantiation of Docker, which are now running an Amazon Web Service (AWS) as reflected in Figure 64. This capability was initially rolled-out in mid-February 2018. The Docker to deploy OpenMBEE is evolving and details of the process have been provided to our ARDEC sponsor. An important use case is in a new era where the government plans to move to digital engineering and model-based acquisition, the government can provide models and generated "specifications" from the models using DocGen; in addition, the environment that was used to produce the models can also be provided using Docker.

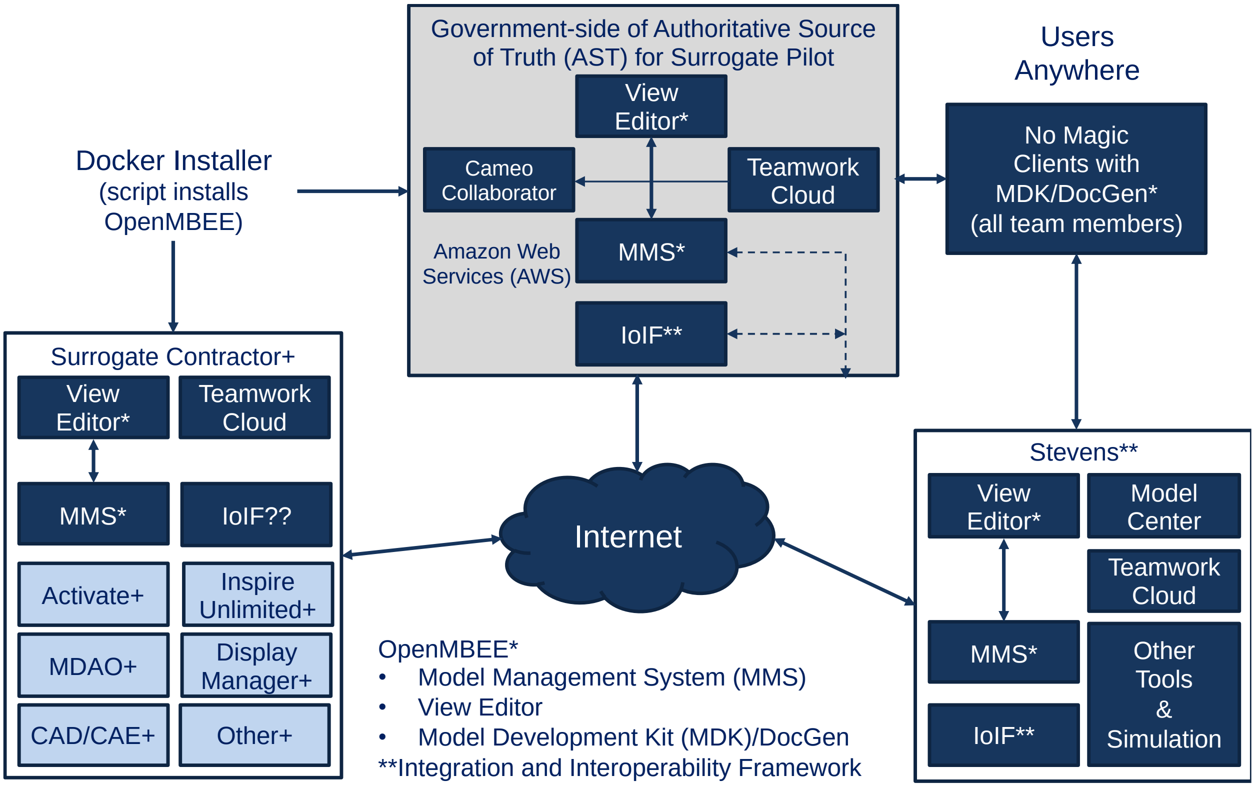


Figure 64. Elements of Authoritative Source of Truth – Including IoIF, MagicDraw, MMS, View Editor, Teamwork Cloud and Various Software Tools

12.2 INTEGRATED MODELING ENVIRONMENTS

This section reviews some of the most advanced tool integrations and integrated modeling environments that have been developed by NASA/JPL [67] [11], the DARPA META projects [9] [8], Engineered Resilient Systems [93], Airbus [88], and generalization of commercial and industry integrated modeling environments. We cannot discuss much about industry integrated modeling environments, but we know that industry organizations are part of the Open Collaboration Group for MBSE and OpenMBEE [150]. We know they are using OpenMBEE. We participate in this Open Collaboration Group both as part of the leadership team and as committers (e.g., Docker). We look to take advantages of the OpenMBEE and other open source tools to demonstrate the art-of-the-possible in our research.

We review the contributions that have been removed from the report, because they have been de-scoped from the current research efforts. However, the research analysis can be found in the Phase I RT-168 Final Technical report [28]. They include:

- Performed PTC Windchill [196] analysis for the Army under the SERC RT-152 [118], which was expanded to identifying capabilities for the AVCE iMBE concept.
- Performed initial analysis of the Syndeia by Intercax [186] through demonstrations with our ARDEC sponsor
    - Syndeia is a software platform for MCE to enable engineering teams to collaboratively develop and manage a system model, and provides a means to combine a system architecture model defined in languages such as SysML with models in other MBE

    - domain, including PLM (e.g. Teamcenter, Windchill), CAD (e.g. NX, Creo), Application Lifecycle Management (ALM) (e.g. GitHub), Project Management (e.g. JIRA), Requirements Management (e.g. DOORS-NG), Simulations (e.g. Mathematica and MATLAB/Simulink), Databases (e.g. MySQL), and other data sources (e.g. Excel)
    - We still think there are opportunities with Syndeia, but do not have the resources to address this task.
- Attended a presentation related to a workbench platform for integrating tools on that supported the ARDEC/ANSYS Developed Analysis Preprocessing Tool (ADAPT).
    - We think there are opportunities to apply the ANSYS workbench and are working to obtain academic licenses.
- Attended demonstration of the integration between OpenMBEE, No Magic and Siemen's Teamcenter.
- Attended demonstrations of the integrations between OpenMBEE and No Magic products.
- Attended demonstrations of the integrations between OpenMBEE and Tom Sawyer visualization products.

12.3 CANONICAL REFERENCE ARCHITECTURE OF AN INTEGRATED MCE ENVIRONMENT

Recalling that a critical element of the first year of this research is to understand the requirements for AVCE iMBE, we believe that the RT-141 final report [26] generalized capabilities heard by many organizations [8] [9] [11] [51] [67] [93] [102] [157] and characterizes a canonical reference architecture of an Integrated MCE Environment, as shown in Figure 65. The following sub-sections discuss various elements from the canonical reference architecture for an integrated MCE environment. 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 Truth (SST), which may better be characterized as an Authoritative Source of Truth (AST) as reflected in Figure 64
- 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 [163]
- MDAO layer provides for problem and design space exploration of
    - Physics-based models
    - Integrity-based models
    - Cost and scheduling models
    - Risk models
    - Various "illities" models
    - Including surrogates and components
- Enabled by High Performance Computing (HPC)
- Semantically rich linkages between data and information in the SST provides for continuous workflow orchestration – enabled by HPC
- Document generation is enabled by
    - Semantically rich links to information in the SST
    - 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)

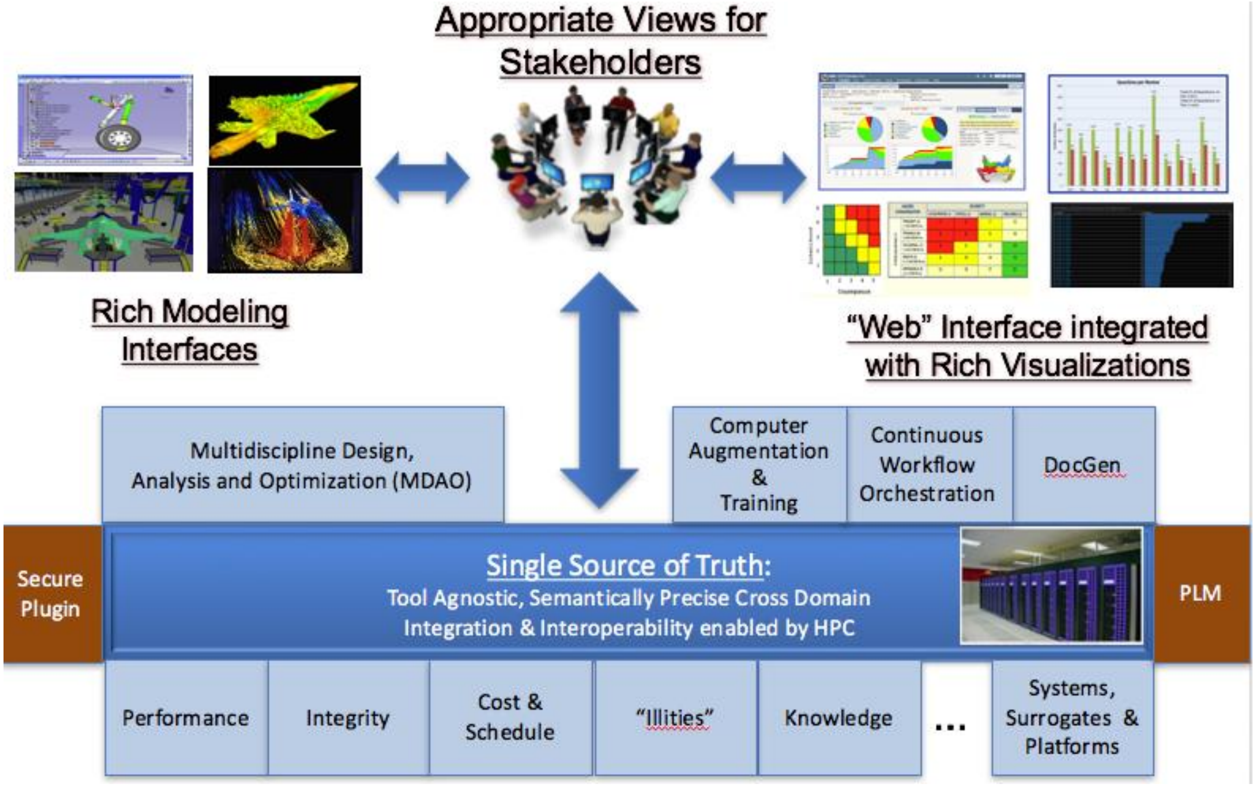


Figure 65. Integrated Environment for Iterative Tradespace Analysis of Problem and Design Space

Therefore, the elaboration of the subtasks as described in Section 12 come from insights gained in discussions from over thirty organizations, related SERC analyses, and new research findings.

12.4 THE CHALLENGE OF DYNAMIC NATURE OF TOOL INTEGRATION

We have discussed the use of SWT as a means for interoperability, and our ARDEC sponsors understand our views on this topic; this section provides additional justification on this view assembled during this research. Tool integrations are dynamic consequences of customer requirements, especially when trying to address the cross-domain integrations as described in Section 3.2. Tool integration are not simply statically putting a certain set of tools together. Depending on the varying needs of tasks from particular stakeholders, the types of tools needed, their execution sequences, the interdependencies of data flow among them vary from case to case. In addition, the problem often gets worse when attempting to maintain an integration for different versions of tools. Figure 66 illustrates the dynamic nature of tool integration [176].

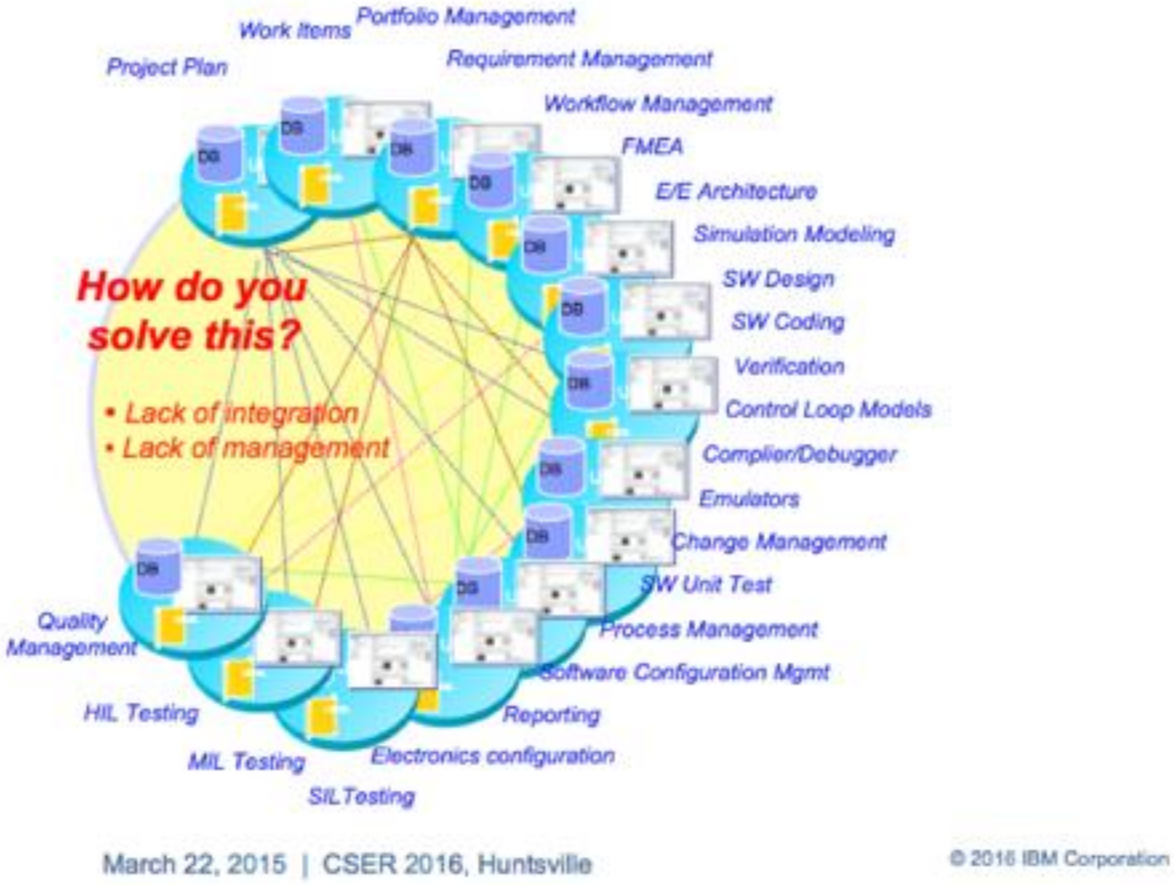


Figure 66. Coordination Across Tools Based on User Story

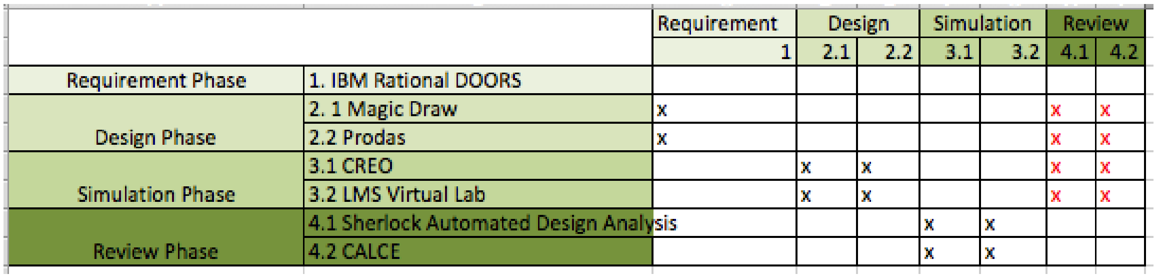
12.4.1 ANALYZING TOOL INTEGRATIONS

We initially started this task looking at the application of a multitude of tools used in modern product development, aligning mostly with MCE. Complexity arises as the volume of the needed tool set and their inter-dependencies increase. The design structure matrix (DSM) has been demonstrated to be very helpful for representing and analyzing the architecture of an individual system, such as a product, a process, and an organization [59]. A DSM is often a two-dimensional matrix representation of the structural or functional interrelationships of objects, tasks or teams. Synonyms for DSM can be N2-Diagram ("N-squared"), and Dependency Structure Matrix. Types of DSM found in use include object-based, team-based, parameter-based, task-based, software module-based, and tool-based.

In this use case, we initially planned to explore the potential of DSM in addressing challenges associated with integrating various tools in product development. However, our researcher did not have detailed insights into many of these tools, several of which have been created by ARDEC to serve very special purposes in their analysis and designs. ARDEC in the second working session discussed some of these integrations, but we are not including those details in this report due to the labeling on the presentation material; we are not distributing this material either. Therefore, we have concluded that in order to attempt to do the DSM analysis, we would have need significant support from ARDEC or other experts that can discuss how they use the tools. Therefore, this section describes why and how we would attempt to perform this type of analysis.

12.4.2 THE OVERALL DSM FOR TOOL INTEGRATION

Given a comprehensive set of available tools that may be potentially used in different phases of product development. We can construct a DSM to represent their relationships. As a toy example shown in Figure 67, the rows and columns can represent available tools, ranked in layers following the temporal order that tools can be used in various phases of product development. Each cell in the matrix can represent the dependency between the tool on the row and the tool on the column. For example, CREO (a 3D CAD software) may use the design blueprint created by the Prodas tool (weapon design tool) for 3D visualization, hence, there exist a dependency from the CREO to Prodas.



		Requirement	Design		Simulation		Review	
		1	2.1	2.2	3.1	3.2	4.1	4.2
Requirement Phase	1. IBM Rational DOORS							
Design Phase	2. 1 Magic Draw	x					x	x
	2.2 Prodas	x					x	x
Simulation Phase	3.1 CREO		x	x			x	x
	3.2 LMS Virtual Lab		x	x			x	x
Review Phase	4.1 Sherlock Automated Design Analysis				x	x		
	4.2 CALCE				x	x		

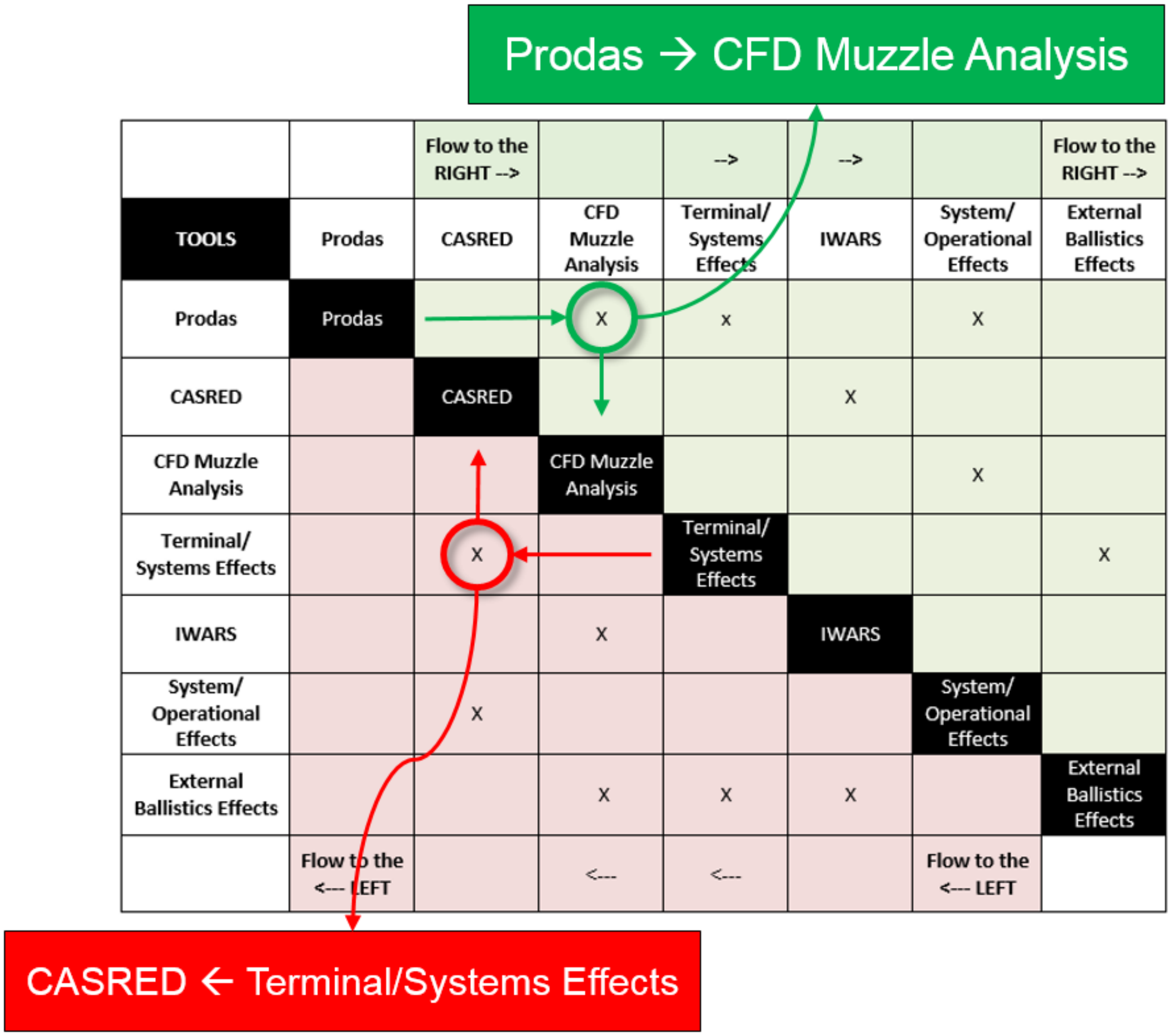
Figure 67. Overall DSM for Tool Integration

In general, the dependencies among tools form a hierarchy, where the later phase tools depend on the prior phase tools. However, there are exceptions, where the design and simulation phase tools can depend on the review phase tools. This is because the review phase tools can generate feedback information, which can lead the product development life cycle to iterate back to re-design and re-simulation.

12.4.3 CAPTURING WORKFLOW INFORMATION USING DESIGN STRUCTURE MATRIX

ARDEC has identified about 85 tools that should be considered as part of various workflows, which cover the entire lifecycle. As shown in Figure 68, they are investigating the use of the DSM concept for capturing information about the numerous workflows that exist at ARDEC.

- Basic question: what tools provide information used by other tools?
- Upper/right portion (Green) - identify sequence from left to right.
- Lower/Left portion (Red) - Identify sequence from right to left.
- Example. Output from Prodas is used as input to CFD Muzzle Analysis.



		Flow to the RIGHT -->		-->	-->		Flow to the RIGHT -->
TOOLS	Prodas	CASRED	CFD Muzzle Analysis	Terminal/ Systems Effects	IWARS	System/ Operational Effects	External Ballistics Effects
Prodas	Prodas		X	x		X	
CASRED		CASRED			X		
CFD Muzzle Analysis			CFD Muzzle Analysis			X	
Terminal/ Systems Effects		X		Terminal/ Systems Effects			X
IWARS			X		IWARS		
System/ Operational Effects		X				System/ Operational Effects	
External Ballistics Effects			X	X	X		External Ballistics Effects
	Flow to the <--- LEFT		<---	<---		Flow to the <--- LEFT	

Figure 68. Example: Output from Terminal/Systems Effects is used as input to CASRED

12.5 DIGITAL ENVIRONMENT AT AIRBUS SPACE

We have discussed the importance of an underlying information model to enable the cross-domain integration of information in a single source of truth [26]. Ralf Hartmann, the Vice President of Enterprise Digitization gave a technically detailed and highly relevant presentation at the NASA/JPL Symposium and Workshop in Jan 2017 [88]. While there were many points, of particular interest was a historical perspective on how they have been assembling a system design engineering environment to cover the entire lifecycle. The representation of the environment as shown in Figure 69 was particularly interesting as it relates to the concept of a semantically rich information; this pertains to the box in the middle call RangeDB Data Management. This is a relatively recent development where they replaced a commercial product with their own infrastructure functionality (i.e., “secret sauce”) that provides a Semantic Data Model for multi-disciplinary Integration as shown in Figure 70. We did discuss this with a person from Airbus at the event, and asked about the strange name (i.e., RangeDB), and he said it was “historical.” This effort confirms why we believe SWT will play a key role to characterize the underlying information model for both ARDEC and NAVAIR, and again reflects positively on the NASA/JPL use of SWT as discussed in Section 3.1.

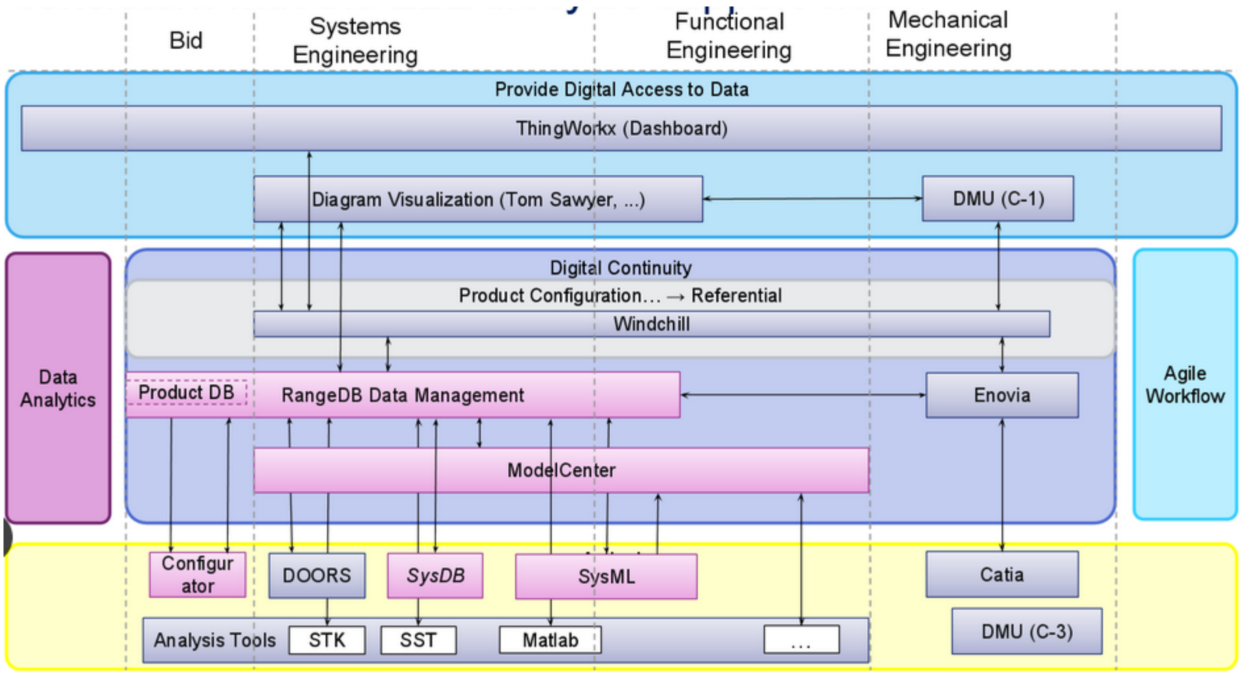


Figure 69. Airbus Digital End-to-End (System & Product) Engineering

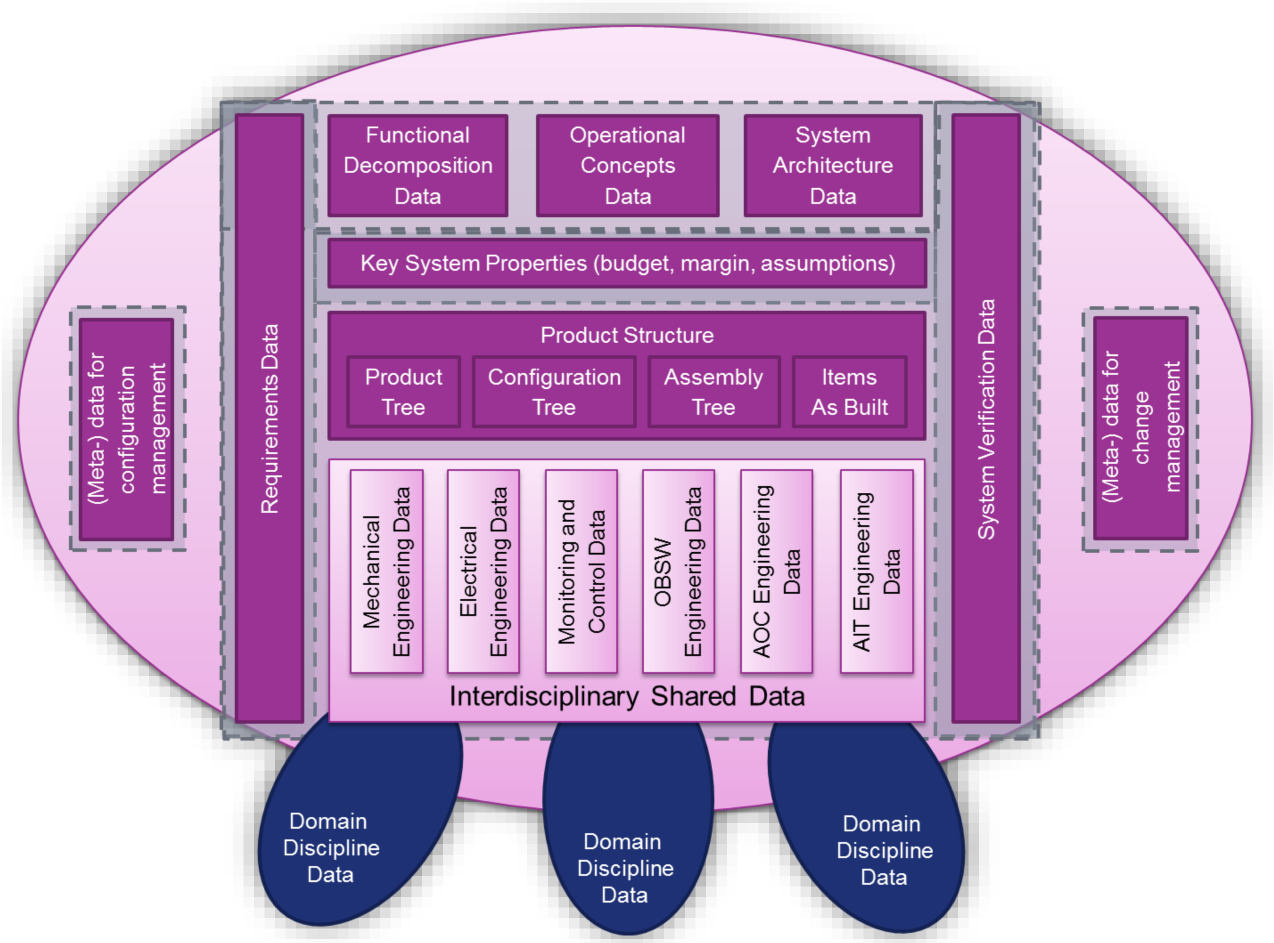


Figure 70. Semantic Data Model for Multi-Disciplinary Integration

Finally, the Hartmann briefing also included an associated roadmap as shown in Figure 71 that was structured in two dimensions:

- Technology clusters
    - Requirement engineering & V&V
    - MBSE and design
    - Engineering data lifecycle management
    - Collaborative engineering
- System engineering technology integration levels
    - Data integration (just connecting data)
    - Semantic integration (identifies rules how to connect and understand data)
    - End-to-end (knowledge management)

The key reflection on this roadmap is acknowledging the increased need to formalize the underlying information model as we move to the right (i.e., future), which can exploit more computational automation enabled by high performance computing.

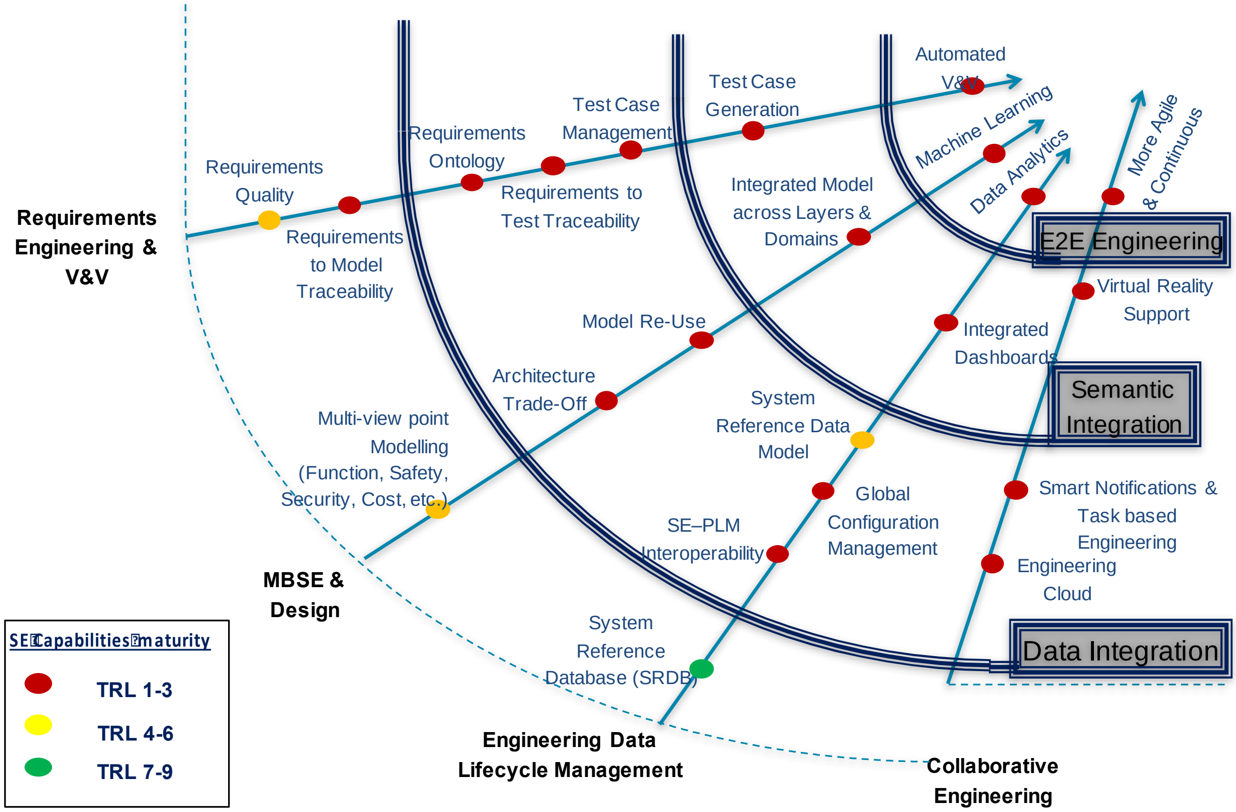


Figure 71. Airbus Roadmap Shown Bands of Digital Engineering Integration

13 RESEARCH SEMANTIC WEB TECHNOLOGIES APPLIED TO AAMODAT (UC10)

This use case relates to UC00 and UC06. As discussed throughout this report, we have moved away from the use of AAMODAT, because of the potential issue of releasability. We have demonstrated accomplishments on our research of the Decision Framework by extending the capabilities in the context of IoIF, which include OpenMBEE, decision ontology and with SWT, SysML, MBSEPak, ModelCenter and some visualization capabilities such as Tableau as shown in Figure 6.

14 ASSESS AVCE IMBE (UC11)

We were requested by ARDEC to provide a peer review of the requirements for AVCE iMBE. While ARDEC has finished the Systems Requirement Review for AVCE iMBE, we asked Rick Dove to join RT-168 research team, because Rick has done some interesting work on the INCOSE’s Agile Systems Engineering Life Cycle Model (ASELCM) project, and specifically, the ASELCM Pattern of Three Concurrent Systems [173]. Agile systems engineering encompasses three nested concurrent systems, depicted in Figure 72 as an iconic pattern. The pattern is the work of Bill Schindel, a principle co-author in the ASELCM case studies. The ASELCM Pattern establishes a set of system reference boundaries. Whether the systems of interest are small or large, human or inanimate, flying through the air or performing business processes.

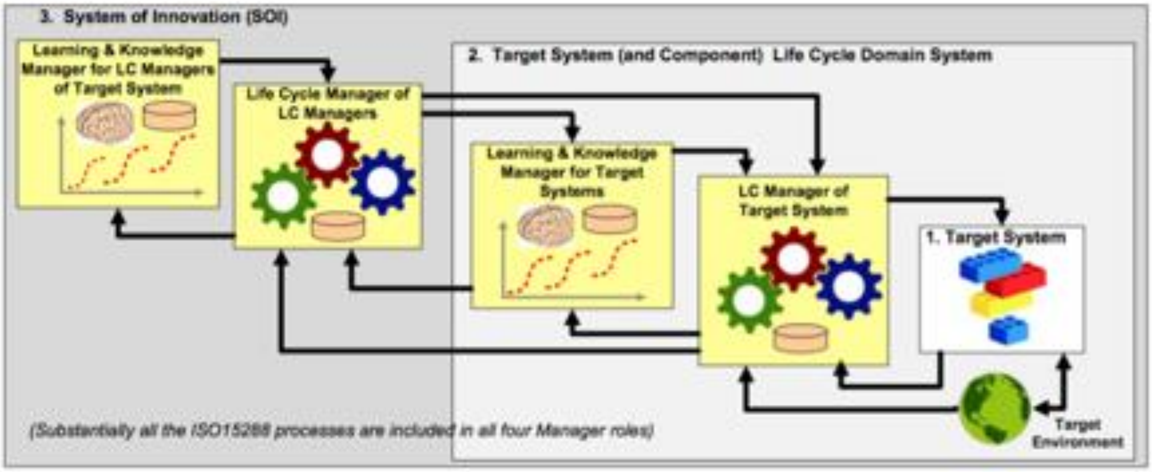


Figure 72. Notional Relationships of Systems 1, 2, and 3 [173]

This ASELCM Pattern particularly refers to three major system reference boundaries, and within those, six subsystem reference boundaries. These are all logical boundaries (defined by the behavior, not the identity, of systems):

- System 1: The Target System, the subject of innovation over managed life cycles of development, deployment, and support.
  - Normally, one would think about the target system as the one that ARDEC would deploy (e.g., fire control, munitions)
  - In this case, however, the target system is AVCE iMBE
- System 2: The Target System Life Cycle Domain System, including the entire external environment of the Target System—everything with which it directly interacts, particularly its operational environment and all systems that manage the life cycle of the Target System. This includes the external environment of the operational target system(s), as well as all the (agile or other) development, production, deployment, support, security, accounting, performance, and configuration management systems that manage System 1.
- System 3: The System of Innovation, which includes System 1 and 2 along with the systems managing (improving, deploying, supporting) the life cycle of System 2. This includes the systems that define, observe, analyze (as in agile software process retrospective), improve and support processes of development, deployment, service, or other managers of System 1. System 1 is contained in System 2, which is contained in System 3. All are (or at least should be) happening simultaneously, effectively an organic complex system motivated by self-preservation to evolve suitably in an uncontrolled operational environment. Think of the arrow-pointed pipes of Figure 72 as a circulatory system.

Rick Dove approached the review of the AVCE iMBE from the Operational Aspects that Enable iMBE. The purpose of Use Case 11 is to identify iMBE lifecycle activities and to identify if there are feasible ways to analyze the problem space, but we need more ARDEC Use-Cases for iMBE. The word Agile in ASELCM came out of need to determine new competitive capabilities. Based on the characterization of AVCE, it does have attributes that reflects on being "agile."

15 SERC RESEARCH SYNERGIES

An early request of ARDEC was for our research team to help them increase awareness and synergies with other organizations. This section discusses some synergies to the ongoing ARDEC research tasks that are briefly mentioned in this report to inform readers of the relationships to these other activities.

15.1 NAVAIR SYSTEMS ENGINEERING TRANSFORMATION THROUGH MODEL CENTRIC ENGINEERING

There are many related research efforts between ARDEC and NAVAIR, as well as other government organization that are working towards and SE transformation using MCE. The synergistic NAVAIR research tasks include RT-170 and RT-195. While the domains and concern are different way, we are working with different and complementary researchers to cross-pollinate the results. This includes:

- Strategies related to MBSE supported by our Georgia Tech collaborators (Dr. Russell Peak, Steven Edwards)
- Approaches to use SWT investigating cross-domain integration, requirements ontologies, Natural Language Processing of requirements, supported by Mary Bone and our University of Maryland collaborators (Dr. Mark Austin, Dr. Leonard Petgna)
- MDAO examples of UAVs
- Instantiations of OpenMBEE using Docker that include IoIF

15.2 RT-176 VERIFICATION AND VALIDATION (V&V) OF SYSTEM BEHAVIOR SPECIFICATIONS

Our NAVAIR sponsor had requested that the SERC RT-176 research task being led by Dr. Kristin Giammarco be aligned with the ongoing research from RT-170 and RT-195, as described in Section 10.1.

15.3 AEROSPACE INDUSTRY ASSOCIATION CONOPS FOR MBSE COLLABORATION

This is a follow-up to the effort completed last year which developed a white paper on the Life Cycle Benefits of Collaborative MBSE Use for Early Requirements Development [3]. This white paper discusses the current state and benefits of MBSE across the entire life cycle and provides proposals for addressing such issues as MBSE Collaborative Framework, Government Data Rights, Intellectual Property, and Life Cycle Effectiveness with MBSE.

The effort for this year involves many of the industry contractors to NAVAIR and DoD. The results should produce a white paper describing a CONOPS for how industry and government can collaborate through MCE/MBSE.

15.4 OPENMBEE AND OPEN COLLABORATION GROUP FOR MBSE

We are now part of the leadership team in the Open Collaboration Group for MBSE that is providing support for adopting and contributing to OpenMBEE [150] with our recent submission of a Docker for installation of OpenMBEE.

15.5 SEMANTIC TECHNOLOGIES FOUNDATION INITIATIVE FOR SYSTEMS ENGINEERING

The NASA/JPL Symposium and Workshop on MBSE had a keynote talk given by Steve Jenkins that was fundamentally based on SWT and a foundational ontology for Systems Engineering developed my NASA/JPL. There were also two breakout sessions on the subject SWT. There was significant attendance at the break out session titled: “Ontologies, Formalisms, & Reasoning” possibly due to the motivation given by Steve Jenkins. In general, there is progress being made in this area and there is significant interest. Dinesh Verma has initiated an effort with the support of Chi Lin, Steve Jenkins and Mark Blackburn to bring a community of people together in an attempt to create and ecosystem on Semantic Technologies for Systems Engineering. Eventually other people will become members and participants, under supervision, to continue to shape and expand the ontologies. The working group has created a charter and mission:

- Charter
    - The Semantic Technologies Foundation Initiative for Systems Engineering is to promote and champion the development and utilization of ontologies and semantic technologies to support system engineering practice, education, and research.
- Mission
    - The mission of the initiative is to collect a suite of interoperable ontologies that are logically well-formed and accurate from both scientific and engineering points of view. The initiative will charter a collective of stakeholders that are committed to collaboration and adherence to shared semantic principles for the advancement of systems engineering. To achieve this, initiative working group participants will voluntarily adhere to and contribute to the development of an evolving set of principles including open use, collaborative development, and non-overlapping and appropriately-scoped content. They will capture and maintain metadata for each ontology to encourage implementation and reuse.

The greatest potential for ARDEC is to bring the NASA/JPL Integrated Model Centric Engineering (IMCE) ontologies [135] into the research.

15.6 DIGITAL ENGINEERING WORKING GROUP

We are also participating in the Digital Engineering Working Group, in which both NAVAIR and ARDEC are participating as discussed in Section 1.4. The Office of Deputy Assistant Secretary of Defense for Systems Engineering (ODASD(SE)) formalized the goals, which are:

- G1. Formalize the development, integration and use of models to inform enterprise and program decision making.
- G2. Provide an enduring authoritative source of truth.
- G3. Incorporate technological innovation to link digital models of the actual system with the physical system in the real world.
- G4. Establish a supporting infrastructure and environment to perform activities, collaborate and communicate across stakeholders.
- G5. Transform a culture and workforce that adopts and supports Digital Engineering (DE) across the lifecycle.

These goals are working toward realizing the benefits that were found in Phase I and identified at a recent Government-Industry DE forum conducted by the SERC and the Office of the Deputy Assistant Secretary of Defense for Systems Engineering. The benefits of a DE transformation are [54]:

- Improved Acquisition – by accepting digital deliverables could improve the governments understanding of a projects status and risk along with allowing them to "validate" the contractor's deliverables.
- Improved Efficiency and Effectiveness – reduce time and effort in the performance of existing tasks using a single source of truth for the system.
- 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. This enables the goal of the DoD to establish a supporting infrastructure and environment to perform activities, collaborate and communicate across stakeholders.
- 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. This enables the DoD goal to formalize the development, integration and use of models to inform enterprise and program decision making through an authoritative source of truth.

The special session on ***Systems Engineering Transformation through Model Centric Engineering Past-Why, Present-What, and Future-How*** held on July 31st at Stevens with our ARDEC and our Office of the Deputy Assistant Secretary of Defense for Systems Engineering sponsors included some other special guest from Digital Warfare Office, Naval Surface Warfare Center, MITRE and Raytheon. We had a breakout session looking at the risk and priorities associated with the mapping future research areas to goals of digital engineering transformation strategy as shown in Figure 5.

15.7 NATIONAL DEFENSE INDUSTRY ASSOCIATION MODELING AND SIMULATION

National Defense Industry Association (NDIA) Modeling and Simulation group is looking at approaches for using digital engineering for competitive down select. We are involved in all of these efforts to further the objectives of our sponsor in August of 2016 [137].

16 PART II SUMMARY

This RT-168 final technical report summarizes the accomplishments for this Phase II research. The report outlines the refinement of the tasks with a mapping to evolving use cases that associate the roles of the various researchers and ARDEC stakeholders to other linked use cases to show a non-exhaustive set of dependencies. These dependencies reflect on cross-domain concerns, where discipline-specific stakeholders will ultimately use different technologies, methods and associated analyses. We think this collective set of use cases that are being researched in the context of various related UAV/UAS operational scenarios and case studies are helping us understand both technology and socio-technical concerns that can provide inputs to operational scenarios and requirements for AVCE iMBE.

This report includes the updates characterizing demonstrations, deliverables, models, tools, configurations, reports and research analyses presented during bi-weekly status meeting, as well as the information presented at six (totaling 11) working sessions and two special deep dive working sessions on ontologies and semantic web technology held at Stevens or Picatinny. We have participated or led virtual events, and presentations or demonstrations, but are not limited to:

- Demonstrations of concepts, technologies and framework to leverage integration and interoperability that provides computationally enabled systems engineering to address the challenges of cross-domain model integration of increasingly complex cyber physical systems.
- Demonstrations of mission and system-of-system engineering analysis for new operational approaches such as graphical CONOPS through mission-level, system-level, and component-level model-centric engineering.
- Creating a Docker for deploying OpenMBEE, Model Management System, View Editor, and Model Development Kit (MDK) DocGen component, where we have developed a number of View and Viewpoint hierarchies for using DocGen, including generation of the “specification” for AVCE iMBE and other models.
- Concept for integrating Graphical CONOPS gaming technology to expose functionality, interfaces, controls, and parametric details that are going to be analyzed using Multidisciplinary Design, Analysis and Optimization (MDAO) at the mission-level.
- Creation of a Decision ontology and SWT application to the Decision Framework and formalization Assessment Flow Diagrams using SysML, MBSEPak, and ModelCenter.
- Illustrate how Decision Framework provides methodological guidance for identifying Key Performance Parameters.
- Facilitated several research synergies both SERC (e.g., NAVAIR and non-SERC (NASA/JPL, commercial) to increases ARDEC’s knowledge and leverage insights and foster synergies from other organizations we have been able to leverage.
- Facilitate the acquisition and application of “high-end” MCE commercial technologies to ensure that the research questions are posed in the context of the most advanced technologies used by government and industry.
- Align ARDEC and NAVAIR research with the DoD Digital Engineering Transformation Strategy.

We will continue to align our research needs with the priorities of our ARDEC sponsors to define specific plans for follow-on research beyond RT-168 Phase II that fundamentally aligns or extends the current set of use cases, but with more integration provided with and through the IoIF including the latest SWT, and an ARDEC-aligned set of ontologies. With the advancement of Digital Engineering technologies, there is the possibility to iterate at the Concept of Operations (CONOPS) and mission level applying alternative analyses of capabilities to address mission-level scenarios driven by continuous evolution of system capabilities that are composed at the mission-level. There are many future research topics that still apply to ARDEC needs that include, but are not limited to:

- Investigate the feasibility to develop Hierarchical Decision Framework that can represent tradespaces of mission scenario alternatives, where mission scenarios reference lower-level Decision Framework constructs for system alternatives that can be composed for a mission-level capability.
- Continue to extend IoIF with ontologies in other domains
  - Extend the OpenMBEE integration with ontologies that are interoperability with the decision ontology
  - Integrate visualization into the IoIF such as the preliminary demonstration for the Decision Framework
  - Provide the prototype software and associated component integration capabilities with the characterization of the configurations to ARDEC
  - Investigate how IoIF support Digital Engineering Collaboration in an Authoritative Source of Truth in order to inform ARDEC about AVCE iMBE environment
  - Investigate the potential for using dynamic visualization of alternative analysis to bi-directionally propagate alternative analyses constraints through IoIF through ontologies
- Develop Interoperable Ontologies to characterize the underlying Information Model of the ARDEC-relevant domains
- Investigate the impacts of SysML 2.0 on system modeling methods and SysML in the context of the capabilities of IoIF, and ontologies for systems engineering
- Visualizations
- Probabilistic/Stochastic Analysis Techniques and their relationships across use cases
- Conduct deep-dive sessions to facilitate more rapid knowledge transfer using hands on use of the capabilities broadly characterized in terms of IoIF
- Conduct working sessions to facilitate knowledge sharing about research findings and accomplishments and understanding sponsor needs in their evolving deployment of digital engineering practices and technologies
- Document methods spanning all relevant modeling efforts
- Other Lifecycle concerns from Digital Engineering
  - Model-based-based Program Management, Contracts + Planning
  - Model-based verification and validation
  - SET- based design
  - SE Methods for AI and Autonomous Systems, with increasing autonomy, adaptation. and intelligent behaviors
  - Computer-aided Design (CAD), behavioral techniques, physics-based/engineering simulations, decision analytics, Computer-aided Manufacturing (CAM), system architecting, prototyping, embedded in a knowledge management environment
  - Investigate the implications of product lifecycle management capabilities on the potential for early methodological guidance that enables faster and more robust capabilities, this can include assessing more “heavy-weight” solutions or hybrid solutions

- Assist and inform on tradespace analysis of commercial-of-the-shelf (COTS), government-of-the-shelf (GOTS), and other tooling capabilities as part of the framework task
- Investigate the use of more socio-technical environments and impacts of web-based social media-based approaches, which has been demonstrated to foster more collaboration in other organizations
- Investigate mission-level simulations that are being integrated with system of systems (SoS) and system simulation that increasingly interoperate with distributed interactive simulation capabilities, augmented virtual reality, and gaming technology
- Investigate methods for measuring the integrity (e.g., accuracy of the predicted margins vs. actual margins after manufacturing) of the outputs from design models, methods for model verification and validation, and methods for certifying and managing the configuration of models that serve as the authoritative source of truth in a digital engineering environment
- Draw on research in uncertainty quantification and model verification
- Investigate Model Composability in the context of mission scenarios assembled from various types of system models to align with the broader goals of digital engineering
- How does model morphological analysis play into model composability
- Investigate how to communicate information that may come from imprecise definitions of variables, differing levels of granularity, differing time bases for simulations, and many other issues
- Multidisciplinary Design, Analysis and Optimization (MDAO) for trade study analyses across use cases, and other technologies such as SmartUQ
- Modeling and simulation of manufacturing and possibly early prototyping

17 ACRONYMS AND ABBREVIATION

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

2D	Two dimensions
3D	Three dimensions
AADL	Architecture Analysis & Design Language
ACAT	Acquisition Category
ACES	Automated Concurrent Engineering System
AFD	Assessment Flow Diagram
AFT	Architecture Framework Tool of NASA/JPL
AGI	Analytical Graphics, Inc.
AGM	Acquisition Guidance Model
AGS	Army Game Studio
ALM	Application Lifecycle Management
AMMODAT	Armament Analytics Multiple Objective Decision Analysis
ANSI	American National Standards Institute
AP233	Application Protocol 233
API	Application Programming Interface
AR	Augmented Reality
ARDEC	Armament Research, Development and Engineering Center
ASELCM	Agile Systems Engineering Life Cycle Model
ASR	Alternative System Review
ATL	ATLAS Transformation Language
AVCE	Armament Virtual Collaboratory Environment
AVSI	Aerospace Vehicle Systems Institute
BDD	SysML Block Definition Diagram
BN	Bayesian Network
BNF	Backus Naur Form
BOM	Bill of Material
BPML	Business Process Modeling Language
C-BML	Coalition Battle Management Language
CAD	Computer-Aided Design
CASE	Computer-Aided Software Engineering
CDR	Critical Design Review
CEO	Chief Executive Officer
CESUN	International Engineering Systems Symposium
CFD	Computational Fluid Dynamic
CGF	Computer Generated Forces
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CONOPS	Concept of Operations
CORBA	Common Object Requesting Broker Architecture
COTS	Commercial Off The Shelf
CPS	Cyber Physical System
CREATE	Computational Research and Engineering for Acquisition Tools and Environments
cUAS	Counter UAS
CWM	Common Warehouse Metamodel
DAA	Data Acquisition and Aggregation layer
DASD	Deputy Assistant Secretary of Defense

dB	Decibel
DBMS	Database Management System
DAG	Defense Acquisition Guidebook
DARPA	Defense Advanced Research Project Agency
DAU	Defense Acquisition University
DCDR	Digital design from Critical Design Review (CDR)
DE	Digital Engineering
DIS	Distributed Interactive Simulation
DL	Descriptive Logic
DLR	DLR Institute of Flight
DoD	Department of Defense
DoDAF	Department of Defense Architectural Framework
DoE	Design of Experiments
DOORS	Requirement Management product
DOORS-NG	DOORS-Next Generation
DSEEP	Distributed Simulation Engineering and Execution Process
DSL	Domain Specific Languages
DSM	Domain Specific Modeling
DSM	Design Structure Matrix
DSML	Domain Specific Modeling Language
E/DRAP	Engineering Data Requirements Agreement Plan
ERP	Enterprise Resource Planning
ESP:HE	ESP: Higher Echelon
ERS	Engineered Resilient Systems
ESP	Early Synthetic Prototype
FAA	Federal Aviation Administration
FEA	Finite Element Analysis
FMEA	Failure Modes and Effects Analysis
FMI	Functional Mockup Interface
FMU	Functional Mockup Unit
FOM	Federation Object Model
GAO	Government Accounting Office
GUI	Graphical User Interface
HLA	High Level Architecture
HPC	High Performance Computing
HPCM	High Performance Computing Modernization
HW	Hardware
I&I	Integration and Interoperability
IBM	International Business Machines
IBD	Internal Block Diagram (SysML)
ICD	Interface Control Document
ICT	Institute for Creative Technologies
ICTB	Integrated Capability Technical Baseline
IDEF0	Icam DEFinition for Function Modeling
IEEE	Institute of Electrical and Electronics Engineers
iMBE	AVCE-Integrated Model-Based Engineering
INCOSE	International Council on Systems Engineering
IPR	Integration Problem Report
IoIF	Interoperability and Integration Frameowk, previously referred to as Integration and Interoperability Framework
IRL	Integration Readiness Level

ISEDM	Integrated Systems Engineering Decision Management
ISEF	Integrated System Engineering Framework developed by Army's TARDEC
ISO	International Organization for Standardization
IT	Information Technology
IWC	Integrated Warfighter Capability
JCIDS	Joint Capabilities Integration and Development System
JEO	Jupiter Europa Orbiter project at NASA/JPL
JSF	Joint Strike Fighter
JPL	Jet Propulsion Laboratory (NASA)
JSON	JavaScript Object Notation
KPP	Key Performance Parameter
KSA	Key System Attributes
LIDAR	Light Detection and Ranging
LOC	Lines of Code
LSL	Lab Streaming Layer
M&S	Modeling and Simulation
MARTE	Modeling and Analysis of Real Time Embedded systems
MATRIXx	Product family for model-based control system design produced by National Instruments; Similar to Simulink
MBE	Model Based Engineering
MBEE	Model Based Engineering Environment
MBSE	Model Based System Engineering
MBT	Model Based Testing
MC/DC	Modified Condition/Decision
MCE	Model Centric engineering
MDA®	Model Driven Architecture®
MDAO	Multidisciplinary Design, Analysis and Optimization
MDD™	Model Driven Development
MDE	Model Driven Engineering
MDK	Model Development Kit – OpenMBEE plugin to MagicDraw
MDSD	Model Driven Software Development
MDSE	Model Driven Software Engineering
MIC	Model Integrated Computing
MMM	Modeling Maturity Model
MMS	Model Management System (part of OpenMBEE)
MoDAF	Ministry of Defence Architectural Framework (United Kingdom)
MOE	Measure of Effectiveness
MOF	Meta Object Facility
MOP	Measure of Performance
MP	Monterey Phoenix
MRL	Mixed Reality Lab
MxRP	Mixed Reality Prototyping
MSDL	Military Scenario Definition Language
MVS	Multiple Virtual Storage
N2	N-squared diagram
NASA	National Aeronautics and Space Administration
NASA/JPL	NASA Jet Propulsion Laboratory
NAVAIR	U.S. Navy Naval Air Systems Command
NAVSEA	U.S. Naval Sea Systems Command
NDA	Non-disclosure Agreement
NDIA	National Defense Industrial Association

NEAR	Naval Enterprise Architecture Repository
NPS	Naval Postgraduate School
NSGA	Non-dominated Sorting Genetic Algorithm
OCL	Object Constraint Language
OMG	Object Management Group
OO	Object oriented
OpenMBEE	Open Model Based Engineering Environment
OpenVSP	Open Vehicle Sketch Pad
OSD	Office of the Secretary of Defense
OSLC	Open Services for Lifecycle Collaboration
OV1	Operational View 1 – type of DoDAF diagram
OWL	Web Ontology Language
PAR	Parametric Block in SysML
PDM	Product Data Management
PDR	Preliminary Design Review
PEA	Post Exercise Analysis
PES	Physical Exchange Specification
PIA	Proprietary Information Agreement
PIM	Platform Independent Model
PLM	Product Lifecycle Management
POR	Program of Record
PRR	Production Readiness Review
PSM	Platform Specific Model
QMU	Quantification of Margins and Uncertainty
RDEC	US Army Research Development and Engineering Center
RDF	Resource Description Framework
RDECOM	US Army Research, Development and Engineering Command
RT	Research Task
RTI	Runtime Infrastructure
RFP	Request for Proposal
RPM	Revolutions Per Minute
RPR FOM	Real-time Platform Reference Federation Object Model
ROI	Return On Investment
SAVI	System Architecture Virtual Integration
SE	System Engineering
SERC	Systems Engineering Research Center
SETR	System Engineering Technical Review
Simulink/Stateflow	Product family for model-based control system produced by The Mathworks
SCR	Software Cost Reduction
SDD	Software Design Document
SE	System Engineering
SFR	System Functional Review
SISO	Simulation Interoperability Standards Organization
SLOC	Software Lines of Code
SME	Subject Matter Expert
SOAP	A protocol for exchanging XML-based messages – originally stood for Simple Object Access Protocol
SoS	System of Systems
Software Factory	Term used by Microsoft
SPARQL	SPARQL Protocol and RDF Query Language
SRR	System Requirements Review

SRS	Software Requirement Specification
SST	Single Source of Truth
SSTT	Single Source of Technical Truth
ST4SE	Semantic Technologies for Systems Engineering
STOVL	Short takeoff and vertical landing
SVR	System Verification Review
SW	Software
SWT	Semantic Web Technology
SysML	System Modeling Language
TARDEC	US Army Tank Automotive Research
TBD	To Be Determined
TRL	Technology Readiness Level
TRR	Test Readiness Review
Turtle	Terse RDF Triple Language
UAV	Unmanned Aerial Vehicle
UAS	Unmanned Aerial System
UC	Use Case
UCAV	Unmanned Combat Air Vehicles
UML	Unified Modeling Language
Unix	An operating system with trademark held by the Open Group
UQ	Uncertainty Quantification
US	United States
USD	US Dollars
USC	University of Southern California
VHDL	Verilog Hardware Description Language
VR	Virtual Reality
V&V	Verification and Validation
XMI	XML Metadata Interchange
XML	eXtensible Markup Language
XSLT	eXtensible Stylesheet Language family (XSL) Transformation
xUML	Executable UML

18 TRADEMARKS

Analysis Server is a registered trademark of Phoenix Integration, Inc.
ANSYS is a registered trademark of Ansys, 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.
CREO is a registered trademark of PTC Corporation.
DOORS is a registered trademark of IBM Corporation.
IBM™ is a trademark of the IBM Corporation
iGrafx is a registered trademark of iGrafx, LCC.
Java™ and J2EE™ are trademark of SUN Microsystems
Java is trademarked by Sun Microsystems, Inc.
LDRA is a registered trademark of Trademark of LDRA Ltd. and Subsidiaries.
Linux is a registered trademark of Linux Mark Institute.
Mathworks, Simulink, and Stateflow are registered trademarks of The Mathworks, Inc.
MagicDraw is a trademark of No Magic, Inc.
MATRIXx is a registered trademark of National Instruments.
Microsoft®, Windows®, Windows NT®, Windows Server® and Windows VistaTM are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. ModelCenter, is a registered trademark of Phoenix Integration, Inc.
Modelica® is a registered trademark of the Modelica Association.
Object Management Group (OMG): OMG's Registered Trademarks include: MDA®, Model Driven Architecture®, UML®, CORBA®, CORBA Academy®, XMI®
OMG's Trademarks include, CWM™, Model Based Application Development™, MDD™, Model Based Development™, Model Based Management™, Model Based Programming™, Model Driven Application Development™, Model Driven Development™
Model Driven Programming™, Model Driven Systems™, OMG Interface Definition Language (IDL)™, Unified Modeling Language™, <<UML>>™
OMG®, MDA®, UML®, MOF®, XMI®, SysML™, BPML™ are registered trademarks or trademarks of the Object Management Group.
Oracle and Java are registered trademarks of Oracle, Inc. and/or its affiliates.
ParaMagic is a registered trademark of InterCAX, Inc.
PHX ModelCenter is a registered trademark of Phoenix Integration, Inc.
PowerPoint is a registered trademark of Microsoft, Inc.
PTD is a registered trademark of PTC Corporation, 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.
Solidworks is and 3DEXPERIENCE, the Compass icon, the 3DS logo, CATIA, SOLIDWORKS, ENOVIA, DELMIA, SIMULIA, GEOVIA, EXALEAD, 3D VIA, 3DSWYM, BIOVIA, NETVIBES, and 3DEXCITE are trademarks or registered trademarks of Dassault Systèmes.

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.
Syndeia is a product of Intercax Corporation.
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.
VT MAK is a product of VT Systems, Inc.
VxWorks is a registered trademark of Wind River Systems, Inc.
Windchill is a registered trademark of PTC, 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.

19 REFERENCES

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

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

[3] Aerospace Industry Association, Life Cycle Benefits of Collaborative MBSE Use for Early Requirements Development, April 2016, http://www.aia-aerospace.org/report/life-cycle-benefits-of-collaborative-mbse-use-for-early-requirements-development/.

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

[5] ARTEMIS-GB-2012-D.46 – Annex 2, 2013. https://ec.europa.eu/research/participants/portal/desktop/en/opportunities/fp7/calls/artemis-2013-1.html

[6] APAN, https://community.apan.org/wg/navair-set/.

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

[8] Bankes, S., D. Challou, D. Cooper, T. Haynes, H. Holloway, P. Pukite, J. Tierno, C. Wentland, META Adaptive, Reflective, Robust Workflow (ARRoW), Phase 1b Final Report, TR-2742, October, 2011.

[9] Bapty, T., S. Neema, J. Scott, Overview of the META Toolchain in the Adaptive Vehicle Make Program, Vanderbilt, ISIS-15-103, 2015.

[10] Bauer, E., Armament Virtual Collaboratory Environment (AVCE) – integrated Model Based Engineering (iMBE), Digital Engineering Working Group, March 28, 2017.

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

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

[13] Becar, J. “A Collaborative Conceptual Aircraft Design Environment for the Design of Small-Scale UAVs in a Multi-University Setting”. 2015.

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

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

[16] Bhatt, D., K. Schloegel, G. Madl, D. Oglesby. Quantifying Error Propagation in Data Flow Models. 20th Annual IEEE International Conference and Workshops on the Engineering of Computer Based Systems. 2013.

[17] Bhattacharya, S., T. Başar, "Game-theoretic analysis of an aerial jamming attack on a UAV communication network," Proceedings of the 2010 American Control Conference,

Baltimore, MD, 2010, pp. 818-823.

[18] Blackburn, M.R., What's Model Driven Engineering (MDE) and How Can it Impact Process, People, Tools and Productivity, Systems and Software Consortium, Technical Report SSCI-2008002-MC, September, 2008 http://www.knowledgebytes.net/downloads/Whats_MDE_and_How_Can_it_Impact_me.pdf.

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

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

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

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

[23] Blackburn, M., J. Dzielski, B. Chell, M. Cilli, S. Hoffenson, R. D. Jones, Applications for Three Research Use Cases in Model Centric Engineering using ModelCenter and MBSEPak, Phoenix Integration Webinar, Feb 7, 2018, https://www.phoenix-int.com/learn-more/webinars/.

[24] Blackburn, M. R., M. Austin, M. Coelho, Modeling and Cross-Domain Dependability Analysis of Cyber-Physical Systems, IEEE International Systems Conference (SysCon), April 2018.

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

[26] Blackburn, M. R., M. Bone, and G. Witus, "Transforming System Engineering through Model-Centric Engineering," Stevens Institute of Technology, SERC-2015-TR-109, Nov. 2015.

[27] Blackburn, M., R., R. Blake, M. Bone, D. Henry, P. Grogan, S. Hoffenson, R. Peak, S. Edwards, M. Austin, L. Petgna, Transforming Systems Engineering through Model-Centric Engineering, SERC-2016-TR-101, January, 2017.

[28] Blackburn, M., R., D. Verma, R. Giffin, R. Blake, M. A. Bone, A. Dawson, R. Dove, J. Dzielski, P. Grogan, S. Hoffenson, E. Hole, R. Jones, B. Kruse, J. McDonald, K. Pochiraju, C. Snyder, L. Xiao, B. Chell, H. Kevadia, K. Batra, L. Ballarinni, D. Henry, P. Montemarano, G. Vesonder, R. Dillon-Merrill, T. Richmond, E. Evangelista, Transforming Systems Engineering through Model-Centric Engineering, Final Technical Report SERC-2017-TR-110, RT-168 (ARDEC), Phase I, August 8, 2017.

[29] Blackburn, M., R., R. Blake, M. Bone, J. Dzielski, R. Giffin, P. Grogan, R. Hathaway, D. Henry, S. Hoffenson, B. Kruse, R. Peak, S. Edwards, M. Ballard, A. Baker, M. Austin, L. Petgna, M. Coelho, Transforming Systems Engineering through Model-Centric Engineering, SERC-2018-TR-103, February 28, 2018.

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

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

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

Systems Engineering Transforming System Engineering through Model-Based Systems Engineering. Technical Report. Systems Engineering Research Center, March 31, 2014. http://www.sercuarc.org/library/view/58.

[33] Blackburn, M., R. Cloutier, G. Witus, E. Hole, M. Bone, Transforming System Engineering through Model-Centric Engineering, SERC-2014-TR-044-2, January 2015.

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

[35] Blackburn, M., P. Denno, Using Semantic Web Technologies for Integrating Domain Specific Modeling and Analytical Tools, Complex Adaptive Systems Conference, Nov. 2015.

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

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

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

[39] Bone, M. A., M. R. Blackburn, D. Rhodes, D. Cohen, J. Guerrero, Transforming Systems Engineering through Digital Engineering, Journal of Defense Modeling and Simulation, 2017.

[40] Bone, M. A., M. Blackburn, G. Witus, H. Eirik, and R. Cloutier, "Model-Centric Engineering," presented at the 2016 Conference on Systems Engineering Research, Huntsville, Alabama, 2016.

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

[42] Brandt, S. A., Stiles, R. J., Bertin, J. J., and Whitford, R., "Introduction to Aeronautics: A Design Perspective", 2nd ed. American Institute of Aeronautics and Astronautics, Reston, Virigina. 2004.

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

[44] Broy, M., M. Feilkas, M. Herrmannsdoerfer, S. Merenda, and D. Ratiu. "Seamless Model-Based Development: From Isolated Tools to Integrated Model Engineering Environments." Proceedings of the IEEE 98, no. 4 (April 2010): 526–45. doi:10.1109/JPROC.2009.2037771.

[45] Business Insider, http://www.businessinsider.com/uav-or-commercial-drone-market-forecast-2015-2.

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

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

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

[49] Cilli, M. Seeking Improved Defense Product Development Success Rates Through Innovations to Trade-Off Analysis Methods, Dissertation, Stevens Institute of Technology, Nov. 2015.

[50] Cilli, M., A New Product Development Trade-Off Analysis Case Study Using a Small UAV

Example, May 2017."

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

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

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

[54] Clifford, M., M. Blackburn, D. Verma, and P. Zimmerman, "Model-Centric Engineering - Insights and Challenges: Primary Takeaways from a Government-Industry Forum," Stevens Institute of Technology, Jul. 2016.

[55] Cloutier, R., D. Hamilton, T. Zigh, P. Korfiatis, B. Esfahbox, P. Zhang, P. Pape, J. O'Brian, and S. Weeks (2013). "Graphical CONOPS Prototype to Demonstrate Emerging Methods, Processes, and Tools at ARDEC," Systems Engineering Research Center, Report SERC-2013-TR-031-2.

[56] Cloutier, Robert & Mary Bone. 2015. MBSE Survey. INCOSE IW 2015. Los Angeles, CA.

[57] Crain, Robert K. 2014. "MBSE without a Process-Based Data Architecture Is Just a Random Set of Characters." In, 1–10. IEEE. doi:10.1109/AERO.2014.6836221.

[58] CRitical SYSTem Engineering AcceLeration, Interoperability Specification (IOS) – V1 D601.021, ARTEMIS-2012-1-332830, 2014.

[59] Danilovic, M., Tyson R. Browning, "Managing complex product development projects with design structure matrices and domain mapping matrices", International Journal of Project Management, Nov 2006.

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

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

[62] DARPA, Wanted: Ideas for protecting against small UAS, http://www.darpa.mil/news-events/2016-08-11

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

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

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

[66] Defense Systems, https://defensesystems.com/articles/2015/08/05/black-dart-counter-uas-exercise.aspx.

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

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

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

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

[71] Department of Defense, Digital Engineering Strategy, Deputy Assistant Secretary of Defense Systems Engineering, www.acq.osd.mil/se.

[72] Docker, https://www.docker.com.

[73] Docker OpenMBEE configuration, https://hub.docker.com/r/openmbeeguest/mms/.

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

[75] DO-178B/ED-12B - Software Considerations in Airborne Systems and Equipment Certification, Radio Technical Corporation for Aeronautics Special Committee 167 (RTCA) December 1992.

[76] Evans, B., Modeling and Simulation Applied in the F-35 Program, Barry Evans Lockheed Martin Aeronautics, 2011.

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

[78] Flager, F., John Haymaker, A Comparison of Multidisciplinary Design, Analysis and Optimization Processes in the Building Construction and Aerospace, Stanford, December 2009.

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

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

[81] Giammarco, K., Practical modeling concepts for engineering emergence in systems of systems, in 12th System of Systems Engineering Conference (SoSE). 2017, IEEE. p. 1-6.

[82] Giammarco, K., K. Giles, Verification and validation of behavior models using lightweight formal methods, Conference on Systems Engineering Research, March 23-25, 2017.

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

[84] Gill, Helen. "From Vision to Reality: Cyber-Physical Systems", HCSS National Workshop on New Research Directions for High Confidence Transportation CPS: Automotive, Aviation, and Rail, November 18-20, 2008.

[85] Gonzales, M., C. Gogu, N. Binaud, C. Espinoza, J. Morlier, and S. Quoniam. Uncertainty quantication in aircraft load calibration. 10th World Congress on Structural and Multidisciplinary Optimization. 2013.

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

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

[88] Hartmann, R., Digital Environment and MBSE Progress at Airbus Space, NASA JPL Symposium and Workshop on Model Based Systems Engineering, January 2017.

[89] Hayhurst, Kelly J., Dan S. Veerhusen, John J. Chilenski, and Leanna K. Rierson. A Practical Tutorial on Modified Condition/Decision Coverage, NASA/TM-2001-210876.

http://techreports.larc.nasa.gov/ltrs/PDF/2001/tm/NASA-2001-tm210876.pdf

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

[91] Herring, M., D. Owens, N. Leveson, M. Ingham, and K. Weiss. Safety-Driven Model-Based System Engineering Methodology. 2007.

[92] Herron, J. Model-Centric Design CAD Design in Aerospace. Retrieved from http://www.findarticles.com/p/articles/mi_hb078/is_199801/aihibm1g16938479, 2006.

[93] Holland, J., Engineered Resilient Systems (ERS) Overview, December 2013.

[94] Hutchinson, J., J. Whittle, M. Rouncefield, S. Kristoffersen, Empirical Assessment of MDE in Industry, Proceedings of the 33rd International Conference on Software Engineering, 2011.

[95] IDEFØ, Computer Systems Laboratory of the National Institute of Standards and Technology (NIST), 1993.

[96] IEEE Std. 1278 (2012). IEEE Standard for Distributed Interactive Simulation.

[97] IEEE Std. 1516 (2010). IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA).

[98] IEEE Std. 1730 (2010). IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP).

[99] International Council on Systems Engineering (INCOSE), “MBSE initiative,” January 2007; https://connect.incose.org/tb/MnT/mbseworkshop/.

[100] ISO/IEC 42010:2007, Systems and Software Engineering -- Architecture Description, 2007.

[101] Jackson, Ethan, and Janos Sztipanovits. “Formalizing the Structural Semantics of Domain-Specific Modeling Languages.” Software & Systems Modeling 8, no. 4 (September 2009): 451–78. doi:10.1007/s10270-008-0105-0.

[102] Jenkins, J. S., N. Rouquette, Semantically-Rigorous systems engineering modeling using SysML and OWL, 5th International Workshop on Systems & Concurrent Engineering for Space Applications, Lisbon, Portugal, October 17-19, 2012.

[103] Jenkins, J.S., NASA/JPL Model-Centric Engineering, Part 3: Foundational Concepts for Building System Models, NASA Academy Online, https://nescacademy.nasa.gov/category/3/sub/17.

[104] Joshi, A., M. P.E. Heimdahl. Model-Based Safety Analysis of Simulink Models Using SCADE Design Verifier. Proc. 24th Digital Avionics Systems Conference. 2005.

[105] http://www.json.org.

[106] Khan, O., G. Dubos, J. Tirona, S. Standley, Model-Based Verification and Validation of the SMAP Uplink Processes, IEEE Aerospace Conference, 2013.

[107] Kim, H., Fried, D., Menegay, P., Connecting SysML Models with Engineering Analyses to Support Multidisciplinary System Development, American Institute of Aeronautics and Astronautics, 2012.

[108] Kim, H., Fried, D., Menegay, P., G. Soremekun, C. Oster, Application of Integrated Modeling and Analysis to Development of Complex Systems, Conference on Systems Engineering Research, 2013; http://dx.doi.org/10.1016/j.procs.2013.01.011.

[109] Korfiatis, P., R. Cloutier, and T. Zigh (2015). “Model-based Concept of Operations Development Using Gaming Simulation: Preliminary Findings.” *Simulation & Gaming,* 46 (5): 471-488. doi: 10.1177/1046878115571290

[110] Kortelainen, J., Semantic Data Model for Multibody System Modelling, Dissertation,

Lappeenranta University of Technology, 2011.

[111] Leveson, N., A New Accident Model for Engineering Safer Systems, Safety Science, Vol. 42, No. 4, April 2004.

[112] Liersch, C. M., K. C. Huber Conceptual Design and Aerodynamic Analyses of a Generic UCAV Configuration, 32nd AIAA Applied Aerodynamics Conference, 16-20 June 2014.

[113] https://www.lua.org/about.html

[114] Martins, Joaquim R. R. A., Andrew B. Lambe. "Multidisciplinary Design Optimization: A Survey of Architectures", AIAA Journal, Vol. 51, No. 9 (2013), pp. 2049-2075.

[115] http://www.mak.com

[116] Mathworks, Matlab, Simulink, Stateflow, https://www.mathworks.com.

[117] Matei, I., C. Bock, SysML Extension for Dynamical System Simulation Tools, National Institute of Standards and Technology, NISTIR 7888, http://dx.doi.org/10.6028/NIST.IR.7888, October 2012, http://nvlpubs.nist.gov/nistpubs/ir/2012/NIST.IR.7888.pdf.

[118] Mattingly, J. D. “Aircraft Engine Design”. AIAA education series. American Institute of Aeronautics & Astronautics. 2002.

[119] McClamroch, N., “Steady Aircraft Flight and Performance. Steady Aircraft Flight and Performance”. Princeton University Press. 2011.

[120] McCormick, B. W., “Aerodynamics, aeronautics, and flight mechanics”. Wiley, New York. 1995.

[121] McDonald, J., M. Kerman, Electronic Product Data Management (ePDM) MPTs to Improve Design for Producibility, Reliability, Availability, Maintainability, and Sustainability, Systems Engineering Research Center, Technical Report SERC-2016-TR-113, 30 September 2016.

[122] McFarland, J., Uncertainty Analysis For Computer Simulations Through Validation And Calibration, Dissertation, Vanderbilt University, May 2008.

[123] McFarland, J., Sankaran Mahadevan, Vicente Romero, Laura Swiler, Calibration and Uncertainty Analysis for Computer Simulations with Multivariate Output, AIAA, October, 2007.

[124] McKelvin, Jr., Mark, and Alejandro Jimenez. “Specification and Design of Electrical Flight System Architectures with SysML.” American Institute of Aeronautics and Astronautics, 2012. doi:10.2514/6.2012-2534.

[125] McKnight, R., Using MBSE Tools to Build a Model of Models, Stevens Institute of Technology, SYS800, Summer 2015.

[126] MDK plugin with DocGen, https://github.com/Open-MBEE/mvn-repo.

[127] MIL-HDBK-516C, Department Of Defense Handbook: Airworthiness Certification Criteria, December 12, 2014.

[128] Model Based Enterprise, http://model-based-enterprise.org/.

[129] Monterey Phoenix Analyzer on Firebird: http://firebird.nps.edu.

[130] Moses, A., M. J. Rutherford and K. P. Valavanis, "Radar-based detection and identification for miniature air vehicles," *2011 IEEE International Conference on Control Applications (CCA)*, Denver, CO, 2011, pp. 933-940.

[131] Mostashari, S.A. McComb, D.M. Kennedy, R. Cloutier, and P. Korfiatis (2012). “Developing a Stakeholder-Assisted Agile CONOPS Development Process.” Systems Engineering, 15 (1): 1-13. doi: 10.1002/sys.20190.

[132] Murray, Brian T., Alessandro Pinto, Randy Skelding, Olivier L. de Weck, Haifeng Zhu, Sujit Nair, Narek Shougarian, Kaushik Sinha, Shaunak Bodardikar, and Larry Zeidner. META II

Complex Systems Design and Analysis (CODA), 2011.

[133] NAOMI Project, Lockheed Martin Advanced Technology Laboratories; http://www.atl.external.lmco.com/programs/STI/programs/program1.php#experimentalinfrastructure, 2013.

[134] NASA/JPL, An Integrated Model Centric Engineering (IMCE) Reference Architecture for a Model Based Engineering Environment (MBEE), NASA/JPL, Sept, 2014.

[135] NASA/JPL Integrated Model Centric Engineering ontologies, https://github.com/JPL-IMCE.

[136] National Academy of Science Interim report on Cyber-Physical Systems Education, 2015.

[137] National Defense Industry Association Modeling and Simulation Committee, Final Report on the Use of Digital Models for Competitive Down-Selection Workshop of 15-16 August 2016, March 2017.

[138] National Institute of Standards and Technology, Foundations for Innovation in Cyber-Physical Systems, Workshop Report, 2013.

[139] NAVAIRINST 13034.1C, NAVAIR Instruction: Flight Clearance Policy For Air Vehicles And Aircraft Systems, September, 28, 2004.

[140] Navy Integration and Interoperability (I&I) Integrated Capability Framework (ICF), Operational Concept Document, Version 2.0, 30 September 2013.

[141] Newcomer, J. T., SANDIA REPORT, SAND2012-7912 Unlimited Release Printed September 2012, A New Approach to Quantification of Margins and Uncertainties for Physical Simulation Data. (http://prod.sandia.gov/techlib/access-control.cgi/2012/127912.pdf).

[142] Nixon, D. W., Flight Control Law Development for the F-35 Joint Strike Fighter, October 5, 2004.

[143] No Magic, MagicDraw, https://www.nomagic.com.

[144] Oberkampf, William Louis, Timothy Guy Trucano, and Martin M. Pilch, Predictive Capability Maturity Model for Computational Modeling and Simulation, October 1, 2007, http://www.osti.gov/servlets/purl/976951-meC28s/.

[145] Object Management Group, MBSE Wiki, Ontology Action Team, http://www.omgwiki.org/MBSE/doku.php?id=mbse:ontology, 2014.

[146] Object Management Group, XML Metadata Interchange (XMI), Version, 2.4.2, April 2014, http://www.omg.org/spec/XMI/2.4.2.

[147] Object Management Group. OMG Unified Modeling LanguageTM (OMG UML), Superstructure. 2011. Version 2.4.1. Available from: http://www.omg.org/spec/UML/2.4.1/Superstructure/PDF.

[148] Object Management Group. OMG Systems Modeling Language (OMG SysMLTM). 2012. Version1.3. Available from: http://www.omg.org/spec/SysML/1.3/PDF.

[149] OpenMBEE, http://www.openmbee.org.

[150] https://open-mbee.github.io.

[151] http://openvsp.org.

[152] Papadopoulos, Y., D. Parker, C. Grant. A Method and Tool Support for Model-based Semi-automated Failure Modes and Effects Analysis of Engineering Designs. Proc. 9th Australian Workshop on Safety Related Programmable Systems. 2004.

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

[154] Peak, R., S. Cimtalay, A. Scott, M. Wilson, B. Aikens, D. Martin, Verification, Validation, and

Accreditation Shortfalls for Modeling and Simulation, ￼Final Technical Report SERC-2011-TR-018, Systems Engineering Research Center, 2011.

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

[156] Phoenix Integration, ModelCenter http://www.phoenix-int.com.

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

[158] Raiffa, H. (1982). The Art and Science of Negotiation. Cambridge, MA: Belknap Press of Harvard University Press.

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

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

[161] Revill, M.B., UAV swarm behavior modeling for early exposure of failure modes. 2016, Naval Postgraduate School: Monterey, CA.

[162] Rhodes, D. H., A. M. Ross, P. Grogan, O. de Weck, ￼Interactive Model-Centric Systems Engineering (IMCSE), Phase One Technical Report SERC-2014-TR-048-1, Systems Engineering Research Center, September 30, 2014.

[163] Ressler, S., What's That 3D Model Doing in my Web Browser, Model-Based Enterprise Summit 2014, http://math.nist.gov/~SRessler/x3dom/revealjs14/mbeNISTtalk.html#/.

[164] Rigling, B., Low-Cost Acoustic Array for Small UAV Detection and Tracking, accessed as: https://www.researchgate.net/profile/Brian_Rigling/publication/224397542_Low-Cost_Acoustic_Array_for_Small_UAV_Detection_and_Tracking/links/5741f6be08ae298602ee277c.pdf.

[165] Rizzo, D., M. R. Blackburn, Use of Bayesian networks for qualification planning: a predictive analysis framework for a technically complex systems engineering problem, Complex Adaptive Systems Conference, November, 2015.

[166] Rodano, M., K. Giammarco, A Formal Method for Evaluation of a Modeled System Architecture Matthew Stevens Institute of Technology, Complex Adaptive Systems Conference, 2013.

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

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

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

[170] SAE ARP4761. Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment. SAE International, December 1996.

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

[172] Schindel, W. D., Failure Analysis: Insights from Model-Based Systems Engineering. Proc.

INCOSE Int’l Symposium. 2010.

[173] Schindel, W. and R. Dove. 2016. Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern.

[174] Proceedings International Symposium. International Council on Systems Engineering. Edinburgh, Scotland, 18-21 July. www.parshift.com/s/160718IS16-IntroToTheAgileSystemsEngineeringLifeCycleMBSEPattern.pdf

[175] Schroeder, C. A., A Study of How Model-Centric Engineering Relates to Time-To-Market and Agility to Accommodate Customer-Required Changes, Dissertation, Indiana State University, 2011.

[176] Shani, U., Engaging Ontologies in MBSE, Conference on System Engineering Research, March 2016.

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

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

[179] SISO Std. 001 (2015). Standard for Real-time Platform Reference Federation Object Model.

[180] SISO Std. 007 (2008). Standard for Military Scenario Definition Language (MSDL).

[181] SISO Std. 011 (2014). Standard for Coalition battle Management Language (C-BML).

[182] Snooke, N., Model-Based Failure Modes and Effects Analysis of Software. Proceedings DX04. 2004.

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

[184] Soldworks, http://www.solidworks.com.

[185] System Engineering Research Center, INCOSE, Stevens, Report Of The Workshop On The Relationship Between Systems Engineering And Software Engineering, Workshop sponsored by Stevens, INCOSE, SERC, June 2014.

[186] http://intercax.com/products/syndeia/.

[187] Tableau, http://www.tableau.com.

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

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

[190] https://unity3d.com.

[191] http://www.vitechcorp.com/products/core.shtml

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

[193] Wade, J., R. Cohen, M. Blackburn, E. Hole, N. Bowen, Systems Engineering of Cyber-Physical Systems Education Program, World Innovation Summit for Education, Nov. 2015.

[194] West, T., A. Pyster, Untangling the Digital Thread: The Challenge and Promise of Model-Based Engineering in Defense Acquisition, INCOSE INSIGHT, Volume 18, Issue 2, pages 45–55, August 2015.

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

[196] http://www.ptc.com/product-lifecycle-management/windchill.

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

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

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

[200] World Wide Web Consortium. RDF Vocabulary Description Language 1.1: RDF Schema, February 2014 https://www.w3.org/TR/rdf-schema/.

[201] World Wide Web Consortium. SPARQL 1.1 Overview, March 2013, http://www.w3.org/TR/sparql11-overview/.

[202] World Wide Web Consortium. Turtle - Terse RDF Triple Language, 28 March 2011, http://www.w3.org/TeamSubmission/turtle/.

[203] Xie, H. Li, X., C. Liu., The Model-Based and Bidirectional Software Failure Mode and Effect Analysis Method. IEEE Intl Conf on Reliability, Maintainability and Safety (ICRMS). 2014.

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

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

[206] Zimmerman, P., T. Gilbert, F. Salvatore, "Digital Engineering Transformation across the Department of Defense" The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, December 20, 2017, 154851291774705. https://doi.org/10.1177/1548512917747050.

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

PART III: APPENDICES OF RESEARCH DETAILS

This appendix provides some additional details provided at the request of our sponsor and provides additional details about research details provided to our sponsor. Each of these sections was created by one or more of the researchers.

A. ONTOLOGY AND SEMANTIC WEB TECHNOLOGY WORKSHOP SUMMARY

Author: Paul T. Grogan

Stevens Institute of Technology hosted an Ontology and Semantic Web Technology Workshop on February 5, 2018, led by Dr. Paul Grogan. This session provided a hands-on introduction and overview of the core technologies underlying the ontology-driven research on RT-168. Semantic web technologies broadly aim to increase the level of conceptual interoperability in systems engineering practice and automate verification, reasoning, and inferencing activities in a system model. The three workshop objectives aimed to: 1) develop practical understanding of foundational concepts and tools for semantic web technologies and ontology modeling, 2) establish a baseline capability to model engineering artifacts with classes, properties, and individuals in Protégé, and 3) learn how and why to build upon existing ontologies including the Basic Formal Ontology (BFO), Information Artifact Ontology (IAO).

As a motivating example, consider an information model for a pickup truck. A manufacturer may publish information such as the vehicle configuration (e.g. cab, box, engine, drive, and transmission) and performance data such as fuel efficiency and suggested retail price. Other entities such as government agencies or consumer advocacy groups may provide additional information such as consumer satisfaction and safety ratings in separate data sources. Data interoperability issues arise when merging data from disparate sets due to differences in labels or definitions, unit systems or datums, or measurement assumptions. This issue is widely apparent in systems engineering activities as disciplinary design groups within or across organizations aim to share design information with each other.

The Conceptual Interoperability Model (Tolk, 2006) describes successive levels of increasing interoperability among models. Level 0 represents no information exchange across models. Technical interoperability (level 1) establishes a communication protocol for data exchange, for example e-mailing unstructured documents, representing the current state-of-the-practice in many systems engineering organizations. Syntactic interoperability (level 2) introduces a common data format to structure information, for example, a standard file format. Current efforts in model-based systems engineering aim to develop and adopt common modeling languages such as SysML to establish syntactic interoperability. Semantic interoperability (level 3) introduces a common understanding of the data meaning, adopting a common set of terms and definitions for data members. The objective of semantic web technologies for systems engineering is to achieve semantic interoperability among models.

The concept of linked data establishes the intellectual foundation for semantic web technologies. Linked data embodies a simple data structure that represents relationships between “triples”: subjects (a thing), predicates (a property), and objects (another thing) as a directed graph edge from the subject to the object, labeled by the predicate. This concept is highly extensible to represent any complex data as a directed graph or network of relationships.

Resource Description Framework (RDF) is a set of World Wide Web Consortium (W3C) standards for linked data. It defines a common data syntax or file format such as extensible markup language (XML), Turtle, JavaScript object notation for linked data (JSON-LD), etc. to represent linked data. Core definitions describe common vocabulary terms for linked data graphs (RDF), extended modeling constructs for RDF Schemas (RDFS), and literals defined in XML Schema Datatypes (XSD). RDF Schemas, in particular, establish constraints on allowable graphs such as assigning types within a class hierarchy for subjects and objects and assigning domains and ranges for properties. However, RDF Schemas are purely descriptive and cannot infer types based on context, a feature of more powerful ontologies.

Web Ontology Language (OWL) is a set of W3C standards related to ontologies. It extends RDFS concepts to a formal language expressed in a standard data syntax such as functional, Manchester, or RDF syntax. Compared to RDFS, OWL makes clearer distinctions between individuals and classes, establishes restrictions on classes such as properties, equivalence, and disjointness, distinguishes between object properties and data properties, and establishes restrictions on object properties such as equivalence, disjointness, inversion, functionality, reflexivity, and transitivity). In contrast to object-oriented modeling which uses a closed-world assumption where everything possible must be explicitly defined, OWL uses an open-world assumption where everything is possible unless it is explicitly denied. This feature allows inferencing and reasoning algorithms to deduce complex logical statements.

Protégé is a free, open-source software tool to edit ontologies developed and distributed by the Stanford Center for Biomedical Informatics Research at Stanford University. It provides a Graphical User Interface (GUI) to transform user inputs into an OWL ontology. Protégé includes formal verification to catch logical errors, reasoning/inferencing capabilities, and exports to multiple file formats. While useful for prototyping an ontology, Protégé is not ideal for storing instance data which is more efficiently managed in a specialized database (triple store) designed to handle large volumes of linked data.

An ontology's primary value increases semantic interoperability among a growing set of heterogeneous users. Correspondingly, one should minimize the development of new ontologies and leverage to the maximum extend the work of others to benefit from the network effects of semantic standards. Ontologies are typically organized in a hierarchical structure with an upper-level to establish generic common knowledge, a middle-level to represent domain-spanning knowledge, and a lower-level to represent domain-specific knowledge (Obrst et al. 2003). Building from a common base makes it easier to map or integrate concepts between ontologies owing to a common world view.

Basic Formal Ontology (BFO) is an existing upper-level ontology that shows promise and potential as a common base on which to extend other ontologies (Arp et al. 2015; Smith et al. 2015). BFO is a small, domain-neutral upper-level ontology that primarily categories entities (anything that exists) as either a continuant (entities that persist, endure, or continue in time) or occurrent (entities that unfold in time). Continuants are further subcategorized as independent material and immaterial entities or dependent on some other continuant. Returning to the example of the pickup truck, for example, the truck itself and its physical components (cab, box, engine, transmission, etc.) are all material entities, while its suggested retail price and fuel efficiency are dependent qualities.

Further extensions to BFO provide greater ontological specificity for certain concepts. For example, the Information Artifact Ontology (IAO) is a BFO-conformant ontology that emphasizes information-based entities common to systems engineering applications. An information artifact is a dependent continuant that carries information about something else. IAO provides a distinction between ground truth and measurements thereof. For example, the price of the truck is a quality; however, a particular quote from a retailer or manufacturer is a measurement of the price using a quantitative scalar measurement datum (i.e. currency) with associated units (e.g. U.S. dollars). Multiple differing price measurements may arise from different data sources, representing inherent fallibility of information artifacts.

As applied to systems engineering organizations, semantic web technologies and ontologies provide the means to increase the level of interoperability in engineering models and, specifically, attach semantic meaning to structured data. Achieving semantic interoperability allows heterogeneous engineering teams to exchange data more easily by establishing clear definitions and mapping rules to translate between alternative representations. However, there remain some significant challenges. The systems engineering community must identify and focus efforts on specific areas subject to semantic differences and bound development efforts to a reasonable level, as developing ontologies is an effort-intensive process on par with other standards development activities. It is unrealistic to expect ontology development to be a top-down process but rather a system-of-systems architecting process benefitting from stable intermediate forms, triaging efforts to elements under control, exerting leverage at the interfaces, and promoting mechanisms and incentives to ensure cooperation (Maier, 1999).

A.1. References

- Tolk, A., 2006. “What comes after the semantic web – PADS implications for the dynamic web,” *20th Workshop on Principles of Advanced and Distributed Simulation*.
- Obrst, L., H. Liu, and R. Wray, 2003. “Ontologies for Corporate Web Application,” *AI Magazine*, 24(3):49-62.
- Arp. R, B. Smith, and A.D. Spear, 2015. *Building Ontologies with Basic Formal Ontology*, MIT Press.
- Smith, B. 2015. “Basic Formal Ontology 2.0 Specification and User’s Guide.” Online: https://github.com/BFO-ontology/BFO
- Maier, M. 1999. “Architecting Principles for Systems-of-Systems,” *Systems Engineering*, 1(4):267-284.

B. DECISION ONTOLOGY

Authors: Thomas Hagedorn and Ian Grosse

The purpose of the decision ontology in the context of the Interoperability and Integration Framework (IoIF) (Figure 73) is to create a domain neutral and tool agnostic set of terms and information model pattern that can be used to represent information relating to decisions. The decision ontology is defined so as to be generally applicable for virtually any type of decision. The ontology thus provides the set of decision specific terms necessary to understand what the essential parts of a decision are and how they relate to one another. Beyond this, it provides a model for mapping data into the ontology and relating that data to the models or observations that it originates from, as well as to the broader understanding of their relation to the decision. The decision ontology does the same with an ontological treatment of models. This helpfully lets models be expressed in a way that is consistent across modeling tools and languages. This consistent treatment in turn allows the expression of relations between the decision and models influencing the decision-making process.

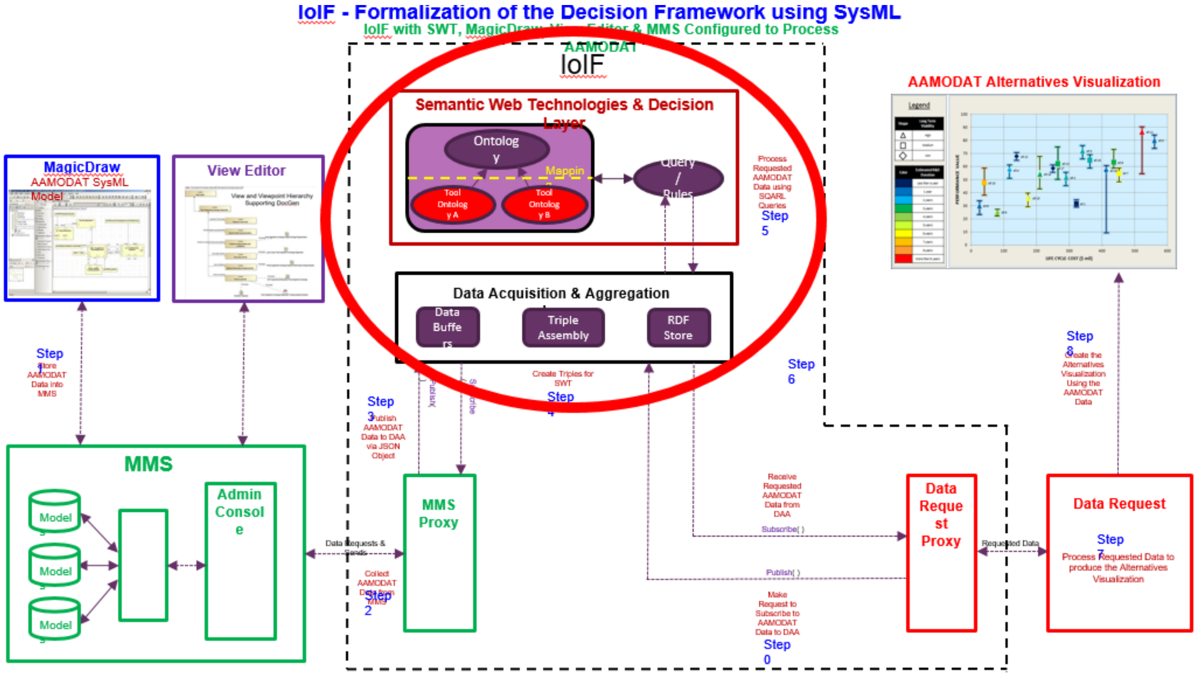


Figure 73. Decision ontology placement in IoIF

In addition to providing terms and an information model pattern for collecting information about decisions, the decision ontology is backed by a number of additional domain ontologies aimed at the engineering domain and at systems engineering, specifically. While the core decision ontology and its dependencies capture decisions, data, and models, the engineering domain ontologies provide terms that can help people and reasoners understand what data means, what is decided about, and what is being modeled. These are primarily aimed at providing context to information captured in the decision ontology. As such, they are mainly focused on providing domain terms in the form of a class hierarchy and defining more specific types of relations that can be used to formulate the information model patterns advanced in the decision ontologies.

B.1. Ontology Design Principles

B.1.1. Prior Work

Several prior efforts have proposed formalized terminology, ontological or otherwise. Though not well suited to this application for various reasoners, this prior work does inform both the terminology of the decision used in the IoIF decision ontology, as well as the basic information model. An early area of application was in computer software development, leading to the development of formal terms for both decisions broadly and for defining various types of stakeholder statements and requirements (Jureta, Mylopoulos et al. 2008, Jureta, Faulkner et al. 2006). These were not, however, implemented in a formal ontology language. Later work resulted in the development of two decision ontologies published around the same time and implemented in the Web Ontology Language (OWL) (Kornyshova, Deneckère 2010, Rockwell, Grosse et al. 2009). The first published by Rockwell et al. (Rockwell, Grosse et al. 2009) focused on the engineering design domain and uses an ontology of engineering analysis models (Grosse, Milton–benoit et al. 2005) in tandem with a set of common terms from engineering decision making. This base is then extended with modular, method-specific terminologies. The second work (Kornyshova, Deneckère 2010) focused on the use of decision making for information systems and implements a fairly detailed information model pattern depicting how the ontology might track decisions. Despite different use domains, the two ontologies are strikingly similar in terminology. They both contain terms such as criteria, requirements, alternatives, and refer to some analytic process used to obtain decisions.

Although helpful, these prior works are nonetheless subject to concerns that ultimately led to the decision not to directly reuse them. Terms are left undefined and without a coherent hierarchical structure. This lack of a distinct is-a taxonomy stems in part from the lack a top-level ontology in either OWL decision ontology. Multiple inheritance is used in several places, causing potentially confusing inferences and query results if used in a practical setting. Where explicit information models are described, relations are often overly constrained, limiting generalizability. These concerns were deemed insurmountable, and so these works were only used as references for potential terms in the newly defined decision ontology,

B.1.2. Ontology Construction

The ontology was designed to maximize its reusability and extensibility to new domains. Thus, several principles were used throughout its development (Arp, Smith et al. 2015). First, a top-level ontology was used to organize and define the terms in the ontology model. Whenever possible, existing previously vetted terminologies were used. Combined, these give the ontology the ability to interoperate with ontologies sharing similar top-level terms and to interface with other ontologies using previously published terminologies. Third, the ontology was designed to be highly modular so as to allow relatively easy selection of only relevant terminologies. For example, the decision ontology itself is composed of three separate, linked ontologies. Where appropriate, these sub-parts can be removed from the ontology as they represent sub-domains, some of which may not be relevant to a given task.

Beyond broad architectural concerns, several previously published principles were used when defining terms and creating the general information model patterns that are used for data capture in the decision ontology. First, the ontology strives to represent reality as realistically as possible (ontological realism). This is most notable in the treatment of data (discussed below) and measurements, but the principle is observed throughout the ontology. Definitions of terms used in several domains are similarly formulated such that they are as general as possible, with sub-terms defined where necessary to indicate domain-specific formulations of terms. Alternatively, additional domain terms are used to formulate a semantic pattern indicating relevant information. For example, a decision certainly can refer to an analysis driven process wherein preference models and system analysis are used to determine a utility-maximizing option. It may alternatively refer to a choice between two alternatives driven by a coin toss. The decision terms are defined such that they are valid in either case but can interact with terms relating to analysis so as to better define the more formal decision case. Similarly, many information model patterns are defined such that a range of individuals (i.e., instances) mapped to domain specific terms can be used at various points without loss of generality.

B.1.3. Ontology Verification

The ontology has been verified using two approaches. The first is a simple test of consistency, which is completed using a description logic reasoner. The Hermit reasoner natively supported in Stanford's Protégé version 5.2 was used to validate the decision ontology. Any reasoner should give an equivalent assessment of consistency. This process is completed both with and without data captured in the ontology. The former ensures consistency of the class hierarchy, while the latter checks the consistency of the information model patterns used in data capture. The second verification approach is one of data capture. It is primarily concerned with whether the ontology terms are sufficient to capture and classify information relevant to a specific application. This can be thought of a verification that the scope of the ontology is sufficient. This was first completed with general decision information read from spreadsheets and subsequently with data parsed directly from a JSON object coming out of MMS. The former case mainly checks the decision terminology, while the latter is an application-specific check. The JSON data also requires that an application level ontology be in place to classify the information.

B.2. Implementation of the IoIF Ontologies

The ontologies are organized hierarchically (Figure 74), such that ontologies are interdependent on one another. In general, all domain level ontologies utilize the upper level, a subset of the middle level, and may have some dependencies on terms defined in other domain ontologies. Even if not directly dependent on one another, the ontologies may interact with one another in semantic patterns provided no axioms in one domain violate restrictions elsewhere in the ontology.

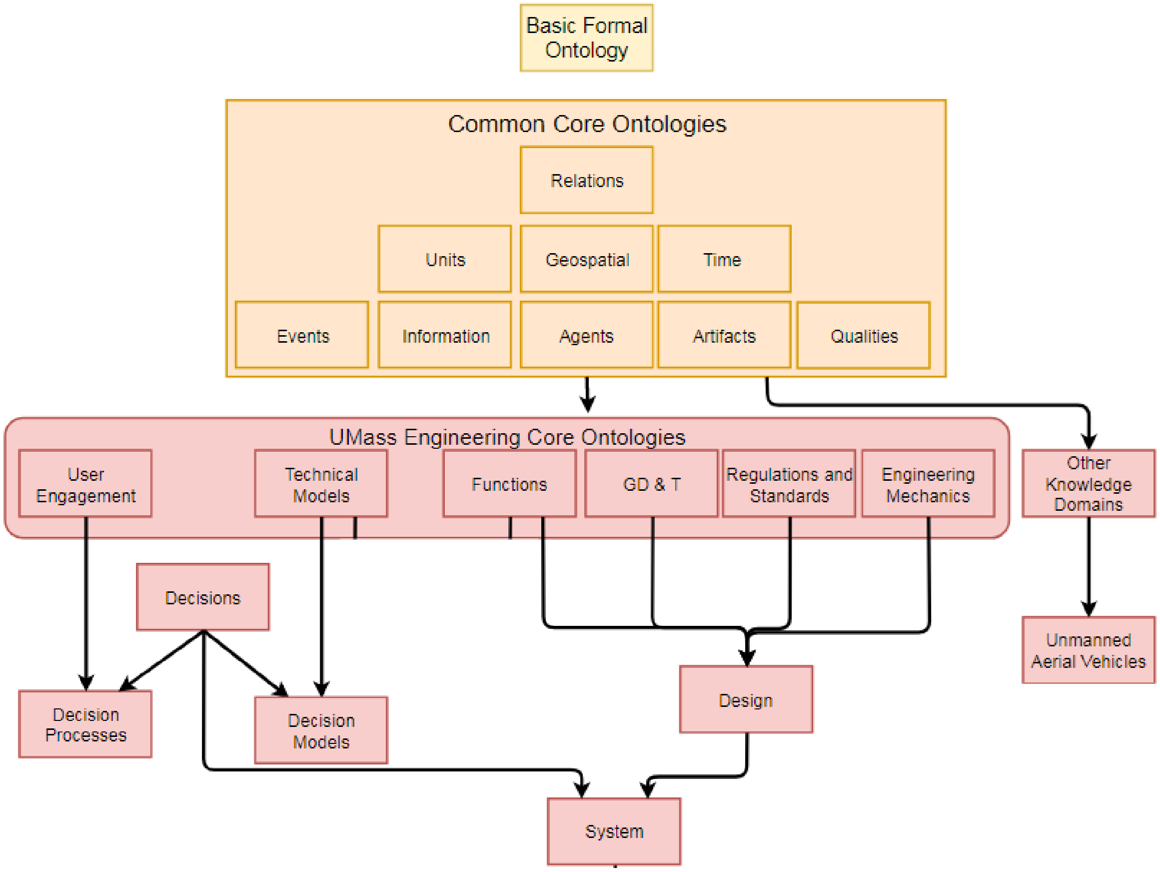


Figure 74. Hierarchical structure of the decision ontology and supporting domain ontologies used in IoIF. Arrows indicate direct dependencies in the form of shared terms

B.2.1. Top Level Ontology

A top-level ontology was used to impose a consistent set of standards for expressing and categorizing information throughout the decision ontology and subsequent linked ontologies for a given application. The Basic Formal Ontology was selected as a top level for the decision ontology and related domain and application ontologies used in the semantic web layer of IoIF. The Basic Formal Ontology (BFO) (Smith, Kumar et al. 2005) was selected based on several criteria. First, it includes only a small, minimal commitment made within the ontology, which means that it does not interfere with domain specific terminology. Second, it is extensively documented (Arp, Smith et al. 2015), both in terms of philosophical viewpoint, best practices, and hard guidelines. This helps with the extensibility of resulting ontologies, as shared principles typically translate to interoperability in ontology engineering. Third, it has been used previously in several domains, meaning that there is a large number of ontologies that can be used to support knowledge capture in BFO (Hagedorn 2018). This allows rapid application of the decision ontology and any tools using it to new knowledge domains.

B.2.2. Core Dependencies

The decision ontology is defined such that it extends, refers to, and or interacts with a number of already defined, BFO conformal ontologies. Beyond BFO as a top-level ontology, a mid-level of relatively common terms and relations is implemented using the Common Core Ontologies[3]. The Common Core Ontologies were developed for use in knowledge representation for intelligence databases. It consists of several, interdependent ontological models addressing both relatively high-level concerns not explicitly modeled in BFO, as well as many terms encountered across several domains. These include an ontology for expressing temporal intervals (Time Ontology), relations between entities (Extended Relations Ontology), and a treatment of different types of information (Information Entity Ontology). These three more fundamental ontologies are used in their entirety in the decision ontology. They are supplemented with additional terms from the remaining Common Core Ontologies. These terms deal with geospatial position, qualities of entities, various types of agent, common types of item, and other common terms.

The decision ontology also utilizes some terms from two additional ontologies developed prior to the effort to create a decision ontology for IoIF. These are an ontology dealing with types of stakeholder outreach and an ontology advancing a set of terms relating to technical models used in engineering. The stakeholder outreach ontology provides terms for describing types of interactions with stakeholders and the types of statements elicited from stakeholders. These terms help capture context within the decision process portion of the decision ontology. The technical model ontology plays a similar role in the decision method portion of the decision ontology largely by providing a general understanding of how value functions and other forms of preference model relate to both the system itself and the stakeholders.

B.3. Domain Dependencies

Given IoIF's intended application area in systems engineering and engineering design and analysis generally, the decision ontology will likely require extension with domain specific terminology. The version currently used in IoIF contains a series of candidate domain ontologies, though should other standards emerge these can be swapped out as needed. Four ontologies are used to describe the engineering domain. These include a function terminology (a refactored, BFO conformal version of the functional basis ontology (Hirtz, Stone et al. 2001, Fernandes, Grosse et al. 2007)), a set of terms relating to dimensions (in effect formalizing dimension specifications used in geometric dimensioning and tolerancing), a set of terms corresponding to human factors design principles (Hagedorn, Krishnamurty et al. 2016), and an ontology formalizing much of the terminology used in the expression of engineering designs (Hagedorn 2018). A set of terms relating to the specific details of systems engineering were also included based on the NASA Jet Propulsion Lab's Integrated Model-Centric Engineering Ontologies (Jenkins 2011). These provide some basis for understanding aspects of a system such as its intended mission and various states that it might hold. Finally, a small set of terms relating to measurement capabilities and aircraft were added to capture the UAS case study for the IoIF specifically.

B.3.1. Application Level Dependencies: SysML and MMS

Application level ontologies extend the general domain knowledge embodied in the decision ontology and related domain dependencies to the specific implementation of IoIF. As SysML is the primary tool used throughout the system and decision modeling process, a set of terms corresponding to the SysML meta-model were added to the ontology. These take the form of terms

---

[3] https://www.cubrc.org/index.php/data-science-and-information-fusion/ontology

corresponding to the types of entity that exist within a SysML model (Slots, Nodes, Actions, etc.) and any tuple fields emerging out of MMS.

The SysML terms were nested within the model ontology used to define the decision ontology. All of these terms are captured as a subclass of a "model primitive," in effect an entity that is part of a model, but not itself meeting the definition of a model. These building blocks typically represent either parts of the entity that is being represented by the model, or actions, relations, and the like that are used to define the model itself. The key concern in this definition is that SysML not be represented in the ontology as the system itself, but as a series of classes, properties, and the like that represent the system or indicate how it is being characterized. Additional subclass axioms can also add in a more explicit representation of SysML definitions in the class term. For example, a SysML abstraction can be indicated as being similar to defining an Is-A relation.

B.3.1.1. Application Properties

Similar to the application terms, the application properties are used to capture information as it is rendered in MMS and the corresponding model. Each term is simply nested within an ontology relation that corresponds to a more general version of application version. This allows the definition in the ontology, and any restrictions placed upon the parent relation to be inherited by the model specific term. A common example of this is the case property restrictions. OWL 2.0 supports several useful property restrictions and property type definitions. Domain terms might be restricted such that their domain (the subject of a triple using the relation) can only be entities having some classification or other traits. Similar restrictions may be placed on the range of the property, which corresponds to the object of a triple, with the property itself being the predicate. These restrictions are inherited in the application level once these terms are asserted to be sub-properties of domain properties. Property type is also inherited.

Property types place additional modifications on how subjects and objects in triples are allowed to relate to one another via a specific property predicate, and what additional relations may be inferred. For example, transitive properties are inferred to have relations such that if A is related to B, and be is related to C, then A is related to C. Functional properties are restricted such that any entity B may be the object of at most one (1) triple having a functional relationship. In both these cases, these relations imply additional information beyond that which is directly asserted. Provided reasoning, the application triples can inherit these inferred relations which can then be queried.

B.3.1.2. Querying Application Level

The application level can be queried using either the native application properties read in from JSON tuples retrieved from MMS, or by using more general ontology terms. The former is obviously application specific but may also reflect nuances that are not captured in more general terms. The latter, however, offers a tool independent approach to retrieving information out of the ontology-linked triple store. One can query independent of tool by adding additional conditions in the query that specifically search based on classifications or relations that are identified via a rdf:SubClassOf or rdf:SubPropertyOf relation to a domain level term. This will return results for any triple matching the query criteria that is mapped to the specified domain terms, irrespective of the application source of that triple (Figure 75).

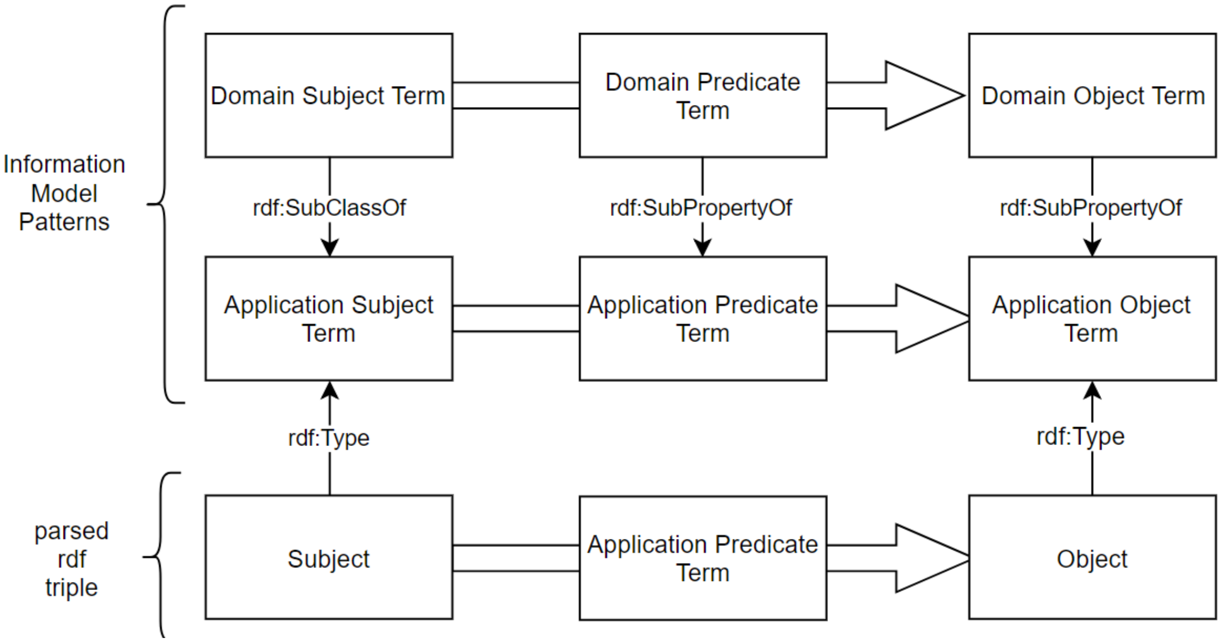


Figure 75. Query patterns that may be used to specify triples (horizontal arrows) that are defined using application level terminology

B.4. Ontology Information Model

B.4.1. General Pattern of Data Capture

The general pattern of data capture used throughout the ontology is defined in the Common Core Ontologies and extended to include key terminology used in decision analysis. Per the principle of ontological realism, the ontology distinguishes between traits (specifically dependent continuants in BFO), the realization of those traits should they need to be realized and the measurement of those traits (for example, speed as a capability of a system versus the measured speed or actual speed of the system during a specific use of the system). Indeed, when dealing with the traits of engineered system, it further distinguishes between three types of data that characterize the system. Artifact designs are directives that indicate the intended traits of some designed system or artifact. Measurements are data items that correspond to specific observations of some entity, such as the measurement of the speed of a projectile with some experiment. Estimates are data items that reflect predicted or anticipated values of some observation. In an engineering context, these commonly arise from models of some entity of interest.

Characterizations of traits are essentially identical irrespective of whether that characterization is measured or estimated. In the case of qualities such as mass or dimensions of a system, these will be measured directly. When rendered as triples the trait of interest would thus be linked directly to the trait of the system via some variant of an "is about" relation. Aboutness within BFO and related ontologies simply indicates that data or other forms of information refer in some way to some other entity. So, in the most general case the corresponding triple might be:

<Data> 'is about' <System Trait>

where <Data Item> is an instance that is some type of "Information Content Entity" and <System Trait> is an instance representing some trait of interest. Were <Data> a measurement, estimate, or directive, one alternative is to replace the 'is about' relation with a more specific one indicating a measurement, estimation, or specification. These are simply sub-properties of the more general 'is about' relation.

In some cases, however, it may be possible to directly measure a trait of a system. Traits that are realizable (such as system capabilities) are innate at any given point in time but are typically characterized via some metric corresponding to a realization of that trait. For example, one might be interested in the speed capability of a system, and so a predictive model might be employed to estimate how fast the system might move under a set of idealized conditions. Alternatively, engineers might consider non-ideal conditions, or characterize speed as an average, maximum, minimum, and so on.

To capture these relations as explicitly as possible, the supporting ontologies add a relation wherein a measurement directly measures one thing but is a metric of another. Thus, the maximum speed under some set of conditions is measured, and this measurement is a metric of the speed capability of the system. This is illustrated in Figure 3.

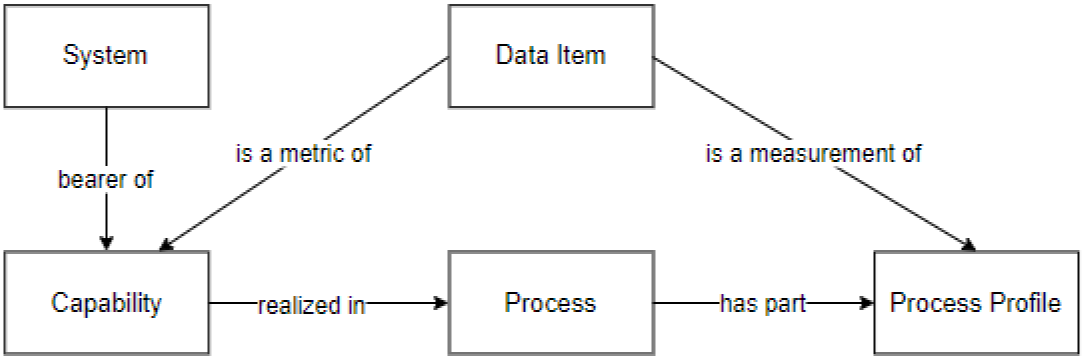


Figure 76. Use of metrics to characterize system traits that are not directly measurable.

B.4.2. Treatment of Models

Models are an important part of multi-attribute decision analysis and are thus given a formal treatment in the decision ontology. The technical model ontology defines a model as a representational information content entity, meaning that it relates to another entity via a "represents" property. When an information entity represents some other entity, it means that there is a one-to-one mapping between the two, wherein one is specifically and only "about" that other entity or that type of entity. Models represent a special case, defined as being one wherein the representation has been deliberately created from a rationale, assumptions, and idealizations using reproducible means to represent known information or predict unknown information about the modeled entity. These predictions are, in fact, estimates that characterize some aspect of the modeled entity, usually some trait of interest.

Models are themselves composed of subparts, a subset of which are model primitives in the model ontology. These model primitives include things such as actions that are undertaken in the model, individual entities representing either real world things or reflecting some internal configuration of the model itself and relationships that are asserted within the context of the model. These are distinct from the actual entities, relationships, etc. that exist in the real world. Instead, they simply reflect how the model represents the real world. Models can also have sub-parts consisting of entire models, equations, and non-model elements that help define the model's internal logic.

Taken together, this treatment of data and models can be used to implement a highly expressive information model pattern that preserves much of the context of any given data point Figure 77. It might distinguish between various types of data or consider what assumptions are implicit in some data point. Alternatively, it might be used to link data to the thing that it is supposed to characterize or to the model it originates from. While much of this information might be known to an expert in a given dataset, implementation in triples mapped to an ontology allows data to be highly searchable. With basic knowledge of the information model, it becomes possible to grab information from virtually any dataset expressed as a graph of triples.

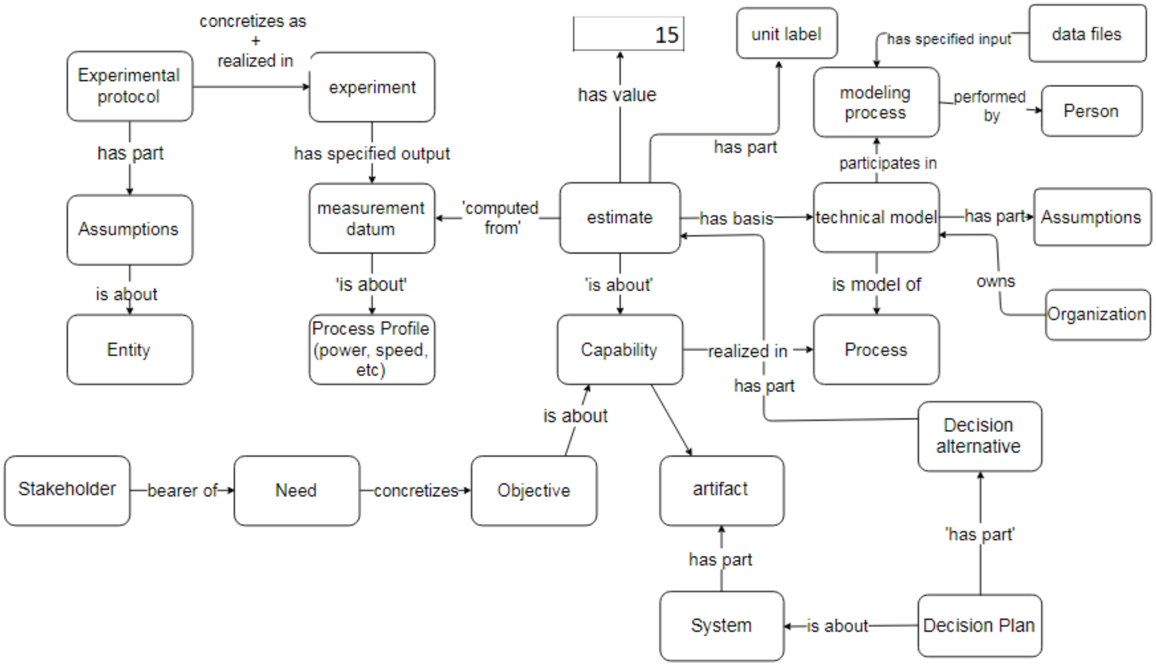


Figure 77. Example of an information model capturing the context of some data point

When considering the representation of models that are captured in MMS and parsed into the IoIF as triples mapped to the ontology, most entities directly parsed will be some form of model primitive. These are basic model sub-parts, which collectively form the entire model. However, provided one has knowledge about what these instances represent, it then becomes possible to construct a representation of the system that is classified as a series of domain terms. The model primitives are then linked to the domain mapped representation of the system. Provided enough prior knowledge about the structure of the incoming model, this could theoretically be done using SPARQL statements at the time of parsing. Irrespective of implementation, the model itself is separate from the representation of the system. However, a key set of model primitives are asserted to represent parts of that system to preserve context and allow querying for specific subsets of the system. Though the exact placement and implementation used to achieve this will vary by tool, the basic approach can be taken for virtually any modeling tool.

In the SysML model used in the UAS case study, for example, much of the model represents internal logic. However, many of the classes explicitly define the UAS system under consideration, its key performance parameters, and specifies a set of measure and value models that directly influence the decision. Provided one knows the general modeling and labelling strategy, it is possible to write queries. For example, one might write queries to access specific classes in a package labelled

“Measure Space” that correspond to the key performance parameters of the system. SPARQL construct queries can be used to define instances corresponding to the traits of the system and specifications detailing how these will be assessed. Additional statements can then define triples asserting that the properties of a class having a specific label represent those measurement specifications. Classes in the package that conform to another pattern, perhaps one including port inputs. Outputs in the model can similarly be located by query and used to construct assertions that said classes represent a sub-model. That sub-model might then be asserted to be the source of subsequent estimates of a key performance parameter. In these and other possible mappings, one need not know specifically what classes represent a key performance parameter. Instead, knowledge about the basic modeling method is used to define an information model pattern, which is selected by a query.

B.4.3. Decision Ontology Design

The decision ontology is defined so as to be generally applicable for virtually any type of decision. Past efforts, as well as decision theory, were consulted to inform the final scope, terminology, and definitions used throughout the ontology. Prior decision ontologies made a useful distinction between a core set of decision terms and method specific ontologies that introduced terms more specifically related to some particular version of a preference model (Rockwell, Grosse et al. 2009). This was judged to be a useful distinction. The specific decision method formalized in IoIF (Cilli 2015) also invests significant effort in the stakeholder engagement process. This process of engagement is largely ignored in prior decision ontologies and is central to understanding the full context of a decision. If not captured in the ontology, much of the basis for a decision cannot be effectively represented in the information model. A preference model might be captured, but the specific stakeholders, interactions, or efforts that go into its creation might be entirely lost.

These considerations point to a need for several views of a decision. The decision itself is a process that unfolds in time. A decision is defined in the ontology as an intentional act wherein some agent or aggregate of agents indicates a preference towards an option from some set of alternatives. This is sufficiently broad as to encompass everything from essentially arbitrary decisions to highly systematic, model-based decision-making processes. A decision specification is a directive resulting from a process. This is, in effect, the information indicating that a decision has occurred and what that decision is. A decision plan specification is a separate directive that indicates how one intends to make a decision. This might detail with whom one will confer, what methods will be used, various requirements, objectives, analytic plans, and so forth.

Thus, the decision ontology is split into three main modules, each having its own namespace. The Decision Process Ontology deals with the actions undertaken in the process of making a decision. This includes the stakeholder outreach terminology, as well as terms to indicate the use of models and to capture the decision event. It also includes terms to identify the roles various stakeholders bear during a decision process. The Decision Method Ontology provides terms relating to how a decision is made. This has two main parts. The first part distinguishes between various types of decision making approach, such as group versus individual decision methods. The second extends general modeling terminology to capture different types of a preference model used in decision-making processes. These terms express the multi-attribute value functions used in many decision-making methods and link these value functions to the specific objectives, metrics, and preferences they ultimately describe.

The core decision ontology provides a set of key terms for decisions generally, irrespective of how they are made. It contains the core decision, decision specification, and decision plan specification terms. It also defines preferences and relations for how preferences relate to the traits of each possible alternative. The basic semantic pattern used in the ontology is that stakeholders possess preferences, and these preferences are ultimately directed towards other entities. In the case of a system, these might be traits possessed by the system (Figure 4). Stakeholders, moreover, possess various needs, which ultimately form the basis of requirements. These requirements are then part of a larger objective specification. Those objectives are then 'about' the system trait, while the requirements are specifications for the system trait.

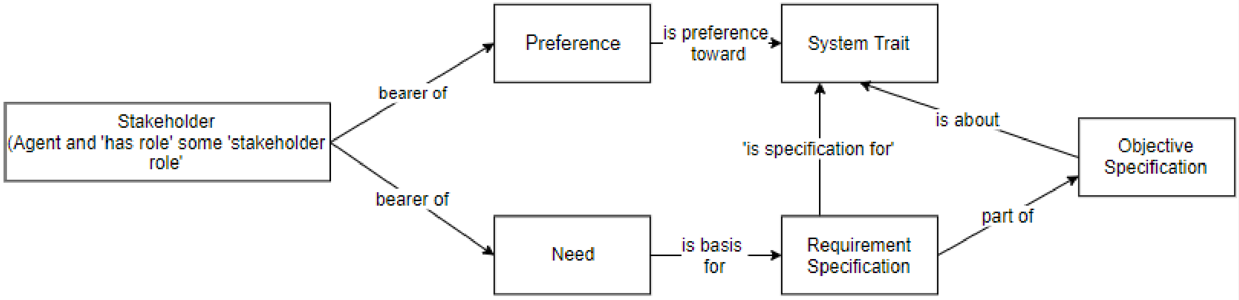


Figure 78. Stakeholders and Preferences

Requirement specifications behave in a manner inherited from the engineering design ontologies. The requirement has a value specification as part of it. That value specification can then specify a number, string, or other datatype that places a constraint upon the range of permissible measurement values for the system trait.

The decision space is treated differently depending on the specific decision. In an engineering design context, alternatives might be a set of design specifications having different sets of predicted behavior. In another context the decision might choose from a set of real world objects, or perhaps a choice in which one indicates a preference for one trait over another. Thus, a realistic decision ontology must allow the expression of all of these possibilities in the face of this broad possibility over what any given alternative might be. Moreover, the treatment must remain consistent with a broader philosophical view from the top-level ontology. To deal with these issues, the decision ontology treats the state of being an alternative in a decision as a designation that is given to certain entities, with that designation being a part of the broader decision plan specification. So, some instance will represent the designation as an alternative for some specific decision. This information entity will then designate all of the alternatives in that decision, which will be members of the appropriate ontological class. Thus, the ontology preserves an understanding of what the alternatives are, while also allowing alternatives to be flagged and searchable as part of some specific decision. Provided reasoning, one can also infer that the decision plan specification is at least in part "about" the things designated as alternatives.

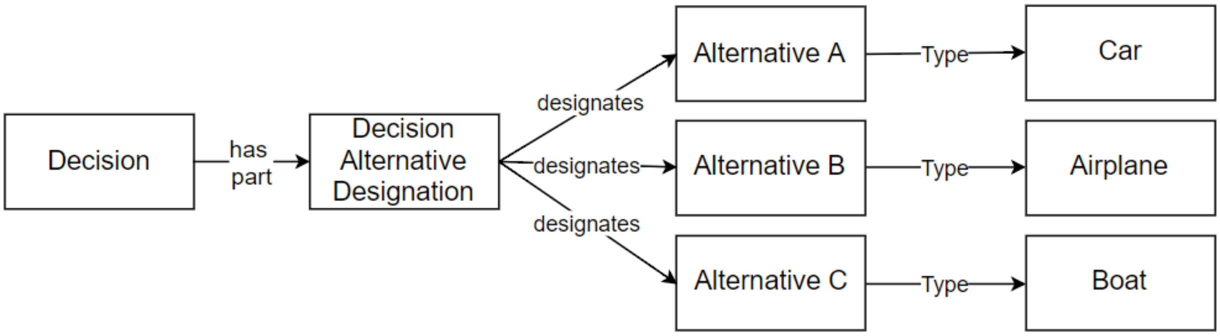


Figure 79. Designating Decision Alternatives

The ontological model of the decision process is straightforward. A plan specification details the exact process by which a decision is made, what the alternatives are, what objectives and/or requirements define the decision, and any models which are going to be used to weight those objectives. The decision itself is a process that realizes the described plan. The decision process results in a selected alternative by the decision maker, as indicated by the "selects" relation. More specific relations can indicate that a decision selects an approval or rejection alternative, or more specific instances wherein the decision maker has specific roles. The outcome of the decision can also be recorded in more detail in the form of a decision specification. This is another information entity that indicates the outcome and ramifications of a decision process and is an output of that decision process. This is preferable to a direct property relation as it allows greater detail when capturing the decision and its ramification.

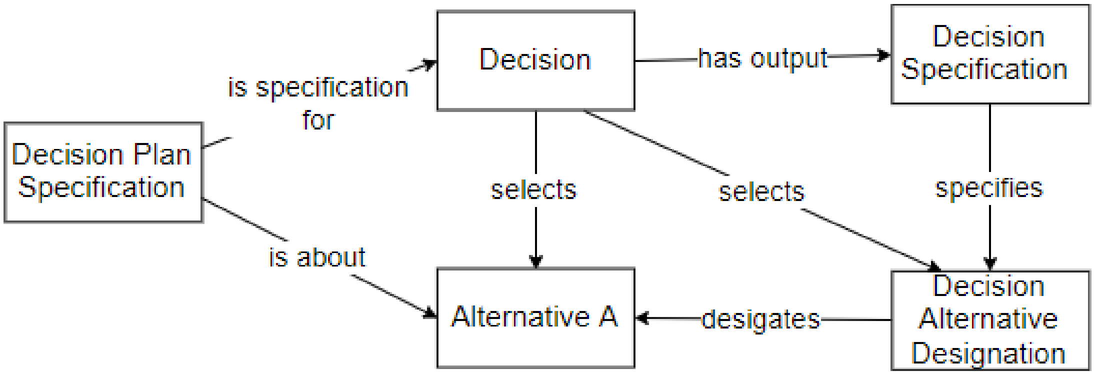


Figure 80. Relation Between Decision Process, Specifications, and Alternatives

The details of the decision method are recorded as part of the plan specification. For a very specific method, much of this might be captured as subclass axioms of an application level class. For example, one might know that while stakeholders will change, value functions will be used, and those functions will be derived using some specific method. More specific information can then be used to fill in the exact details of a specific decision process.

The most up to date version of the decision ontology is freely distributed on the UMass Center for e-Design Github page[4] to promote broader adoption. This page also contains most of the

---

[4] https://github.com/UMassCenterforeDesign/decision-making

dependency ontologies outside of the Common Core[5] and BFO[6] which are hosted elsewhere on Github. As the IoIF develops, the decision ontology is expected to expand and refine its definitions in sub-ontologies or expanded terminology in the method and process ontologies. These changes will be reflected after internal review and published on the online repositories.

B.5. References

ARP, R., SMITH, B. and SPEAR, A.D., 2015. *Building ontologies with basic formal ontology.* Mit Press.

CILLI, M., 2015. *Improving defense acquisition outcomes using an Integrated Systems Engineering Decision Management (ISEDM) approach, .*

FERNANDES, R., GROSSE, I., KRISHNAMURTY, S. and WILEDEN, J., 2007. Design and innovative methodologies in a semantic framework, *ASME 2007 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* 2007, American Society of Mechanical Engineers, pp. 237-247.

GROSSE, I.R., MILTON–BENOIT, J.M. and WILEDEN, J.C., 2005. Ontologies for supporting engineering analysis models. *AIE EDAM,* **19**(01), pp. 1-18.

HAGEDORN, T.J., 2018. *Supporting the Design of Additively Manufactured Medical Devices Using Ontologies.* Retrieved from UMass ScholarWorks, 1 June 2018

HAGEDORN, T.J., KRISHNAMURTY, S. and GROSSE, I.R., 2016. An information model to support user-centered design of medical devices. *Journal of Biomedical Informatics,* **62**, pp. 181-194.

HIRTZ, J.M., STONE, R.B., MCADAMS, D., SZYKMAN, S. and WOOD, K., 2001. Evolving a functional basis for engineering design, *Proceedings of the ASME Design Engineering Technical Conference: DETC2001* 2001.

JENKINS, S., 2011. Introduction to system modeling and ontologies. Jet Propulsion Laboratory, Pasedena, CA.

JURETA, I.J., FAULKNER, S. and SCHOBBENS, P., 2006. A more expressive softgoal conceptualization for quality requirements analysis. *Conceptual Modeling-ER 2006.* Springer, pp. 281-295.

JURETA, I.J., MYLOPOULOS, J. and FAULKNER, S., 2008. Revisiting the core ontology and problem in requirements engineering, *International Requirements Engineering, 2008. RE'08. 16th IEEE* 2008, IEEE, pp. 71-80.

KORNYSHOVA, E. and DENECKÈRE, R., 2010. Decision-making ontology for information system engineering. *Conceptual Modeling–ER 2010,* , pp. 104-117.

ROCKWELL, J., GROSSE, I.R., KRISHNAMURTY, S. and WILEDEN, J.C., 2009. A Decision Support Ontology for collaborative decision making in engineering design, *Collaborative Technologies and Systems, 2009. CTS'09. International Symposium on* 2009, IEEE, pp. 1-9.

---

[5] https://github.com/CommonCoreOntology/CommonCoreOntologies

[6] https://github.com/BFO-ontology/BFO

SMITH, B., KUMAR, A. and BITTNER, T., 2005. Basic formal ontology for bioinformatics. *Journal of Information Systems*, , pp. 1-16.

C. MULTIDISCIPLINARY DESIGN, ANALYSIS AND OPTIMIZATION

Authors: Brian Chell, Steven Hoffenson, Mark Blackburn

A Comparison of Multidisciplinary Design Analysis and Optimization Architectures with an Aircraft Case Study

This paper describes a comparison study of different ways to formulate a multidisciplinary design analysis and optimization (MDAO) problem. Two of these MDAO architectures, multidisciplinary feasible (MDF) and interdisciplinary feasible (IDF), were tested on an aircraft case study. In contrast to many previous MDAO architecture comparison studies, the system being optimized includes simulations. For this case study the aerodynamics discipline is modeled with computational fluid dynamics and the structures discipline is modeled with finite element analysis. The results show that the MDF architecture finds better solutions when it comes to optimality, but it requires more computing resources than does the IDF architecture.

C.1. Introduction

Multidisciplinary Design Analysis and Optimization (MDAO) is a set of methods used to manage optimization problems that involve more than one discipline. It takes into account the interactions among disciplinary models when solving for one or more optimization objectives. Since different subsystems and disciplines will sometimes constrain one another, optimizing in a holistic way using MDAO can be more effective, and sometimes necessary, for achieving optimal solutions. An example of these constraints might be the sizes of the hull and motor in a ship; if the objective is to maximize the ship's speed, then likely the best design would have a narrow hull and a large motor. However, a larger motor requires a wider hull while a smaller motor would allow for a narrower, more hydrodynamic hull. When desired characteristics conflict like this, it is helpful to model the interactions between subsystems in a structured manner. Providing a framework to analyze these interdependencies is one the main benefits of using MDAO. Other benefits include the ability to formally bring together the subsystems that are designed by different teams who are sometimes in different locations. Having the disciplinary models in one computational workflow can help to coordinate their efforts and help them understand their design space.

There are many different ways to formulate an optimization problem that contains multiple disciplines, and the manner in which the problem is formulated can affect both the optimality of the solution as well as the computing resources required to find it. The way in which the disciplinary models are organized and the process in which they are run can vary widely. In this paper these different formulations are referred to as “architectures,” following the lead of Martins and Lambe 2013 in their comprehensive survey. From a high level these architectures are classified first into hierarchical and non-hierarchical, depending on if each “child" subsystem has a “parent" with which it exclusively interacts [2]. Furthermore, the non-hierarchical architectures are broken down into single-level, where one optimizer works the entire problem, and multi-level, where each discipline can have its own optimizer [3]. These two subtypes are also referred to as being monolithic or distributed. While differing architectures have been compared before, there remains work to be done to benchmark these different architectures in order to see how well they perform and how to choose the best for a given optimization problem. A common point made in previous studies is that while simple, analytic problems with low dimensionality have been used to test MDAO architectures, more benchmarking studies need to include computationally intensive problems like simulations with higher dimensionality.

This paper shows the results of a case study where two single-level MDAO architectures, multidisciplinary feasible (MDF) and interdisciplinary feasible (IDF), are compared using a basic fixed-wing aircraft model containing four disciplinary submodels: geometry, aerodynamics, structural mechanics, and performance. Two of these disciplines, aerodynamics and structural mechanics, are modeled using Computational Fluid Dynamics (CFD) and Finite Element Analysis (FEA) simulations. The contribution of this study is to examine how these architectures perform for a simulation-based multidisciplinary problem. Previous studies have largely used analytic equations for their disciplinary models.

These architectures were created with the Phoenix Integration ModelCenter™ MDAO framework [4] which coordinates the disciplinary models. In the following background section, a review of the relevant literature is presented. After that, the methodology section describes the aircraft model development along with the details of the two different architectures used for analysis. Subsequently, the results of these analyses are provided along with a discussion of the findings.

C.2. Background

In this section an overview of MDAO is given. This is followed by a look at MDAO architectures, specifically the MDF and IDF architectures, and a review of the existing research comparing them.

C.2.1. Multidisciplinary Design Analysis and Optimization

MDAO had its beginnings along with computerized optimization in the 1960s, starting with Schmit's influential work using iterative algorithms to minimize the weight of a three bar truss [5] and has been used to successfully optimize many diverse types of systems [6–8]. While the applications of MDAO have been diverse, some of the earliest and most common uses have been to optimize aircraft [9, 10]. MDAO is particularly useful for aeronautical applications since many of the disciplines, in particular aerodynamics and structural mechanics, in aircraft subsystems are highly sensitive to changes in the other subsystems [11].

C.2.2. MDAO Architectures

These early MDAO problems were generally solved by combining all of the disciplinary models into a single optimization problem; this makes the problem similar to an optimization problem with one discipline. Eventually, this method became known as the All-At-Once (AAO) approach. However, formulating an AAO problem becomes more difficult as the complexity increases, if the disciplinary models are using different software tools, and especially if the disciplines are modeled using "black-boxes."

As computing power increased, the deficiencies of AAO limited its usefulness, and more complex MDAO problems were created using novel methods. In 1982, Sobieski broke down complex systems to differentiate disciplinary models and created hierarchical MDAO architectures [12]. From there followed the development of new architectures; Lasdon 1970 postulated methods similar to IDF while optimizing large systems, and then Cramer et al. first 1994 formulated the IDF architecture in order to take advantage of parallel computing. At the same time, they first referred to the “multidisciplinary feasible" approach.

It is important to note the distinction between MDAO architectures and algorithms. A single architecture can be solved using many different algorithms, the selection of which has more to do with the nature of the underlying disciplinary models [15]. The algorithm used for this study and the reasons for selecting it will be described in the methodology section of this paper.

The two MDAO architectures used in this study are MDF and IDF. They are described in the next two subsections.

C.2.2.1. Multidisciplinary Feasible (MDF)

The MDF architecture is perhaps the most straightforward approach at a high level. It takes all of the disciplinary models and combines them into a framework with the outputs of one discipline being directly input into the next; for instance, forces found from an aerodynamic analysis can be input as loads on a structural analysis [14]. Since the outputs of one disciplinary model are directly input into subsequent disciplinary models, the MDF architecture will always remain multidisciplinary feasible. Multidisciplinary feasibility is when both the model and disciplinary submodels reach a feasible point at the end of every optimization iteration. This requires that a multidisciplinary analysis is performed for every iteration which can be computationally costly. The formulation of an MDF problem is shown in equation (1).

$$\begin{aligned} \min_{x} \quad & f\mathbf{x},\mathbf{p} \\ \text{s.t.} \quad & \mathbf{g}\mathbf{x},\mathbf{p} \leq 0 \\ & \mathbf{h}\mathbf{x},\mathbf{p} = 0 \end{aligned} \tag{1}$$

In this formulation, f is the objective function, $\mathbf{g}$ and $\mathbf{h}$ are sets of functions representing, respectively, the inequality and equality constraints, $\mathbf{x}$ is the vector of design variables, and $\mathbf{p}$ is the vector of parameters. The implications of the MDF problem formulation is that, depending on the type of problem, it could require more computer resources. Also, since this requires one analyzer to be used, the benefits of utilizing parallel computing to solve an MDF problem are limited.

C.2.2.2. Interdisciplinary Feasible (IDF)

The IDF architecture separates the disciplinary models, allowing them to use separate analyzers. This requires the addition of a coupling variable constraint, so that the variables passed between each disciplinary model remain consistent across all of the disciplinary models. As opposed to the MDF architecture, it is possible for the individual disciplines to be feasible but for the whole problem to not have multidisciplinary feasibility. This would be the case if the output of one disciplinary model is different from the input to another. One way to overcome this is to add a constraint to the optimization formulation stating that these coupling variables must remain equal. This can be seen in the IDF formulation in equation (2).

$$\begin{aligned} \min_{x,y} \quad & f\mathbf{x},\mathbf{p} \\ \text{s.t.} \quad & \mathbf{g}\mathbf{x},\mathbf{p} \leq 0 \\ & \mathbf{h}\mathbf{x},\mathbf{p} = 0 \\ & \mathbf{y} - \mathbf{a}\mathbf{x},\mathbf{y} = 0 \end{aligned} \tag{2}$$

The notation is the same as in equation (1) except for the coupling variables and constraints. Here $\mathbf{y}$ is the vector of coupling variable inputs and $\mathbf{a}$ are the analysis function outputs that should match up with these inputs. Since these coupling variables need to be added as inputs for the optimizer, it is best to use this architecture when a system has subsystems that are weakly coupled in order to reduce the amount of coupling variables and their corresponding constraints [16]. The compartmentalized nature of IDF is beneficial in that parallel computing can be used, which was the motivation for creating the architecture.

C.2.3. Comparing MDAO Architectures

Although MDAO has been used and studied for many decades, explicitly finding the benefits and drawbacks of differing architectures is more recent. Some of the earliest studies into this topic were in the early to mid 1990’s [17]. Balling and Sobieski 1996 did a survey of six different single and multilevel architectures, finding that the single level architectures generally had better computing times. They also used managerial considerations as ways to evaluate the different architectures, opining that, since work split up into teams will happen concurrently, multilevel approaches would be better.

Another early benchmarking paper compared MDF, IDF, and CO (Collaborative Optimization) [18]. The results were mixed depending on the type of problem being optimized; some of the problems could not be fit into IDF or CO architectures, and MDF consistently converged after fewer function evaluations. These studies used many analytical problems to compare these architectures and noted the difficulty in making these comparisons, as there is no standard way for setting up these architectures. In their survey, Martins and Lambe 2013 mentioned this problem and stated that using MDAO frameworks like ModelCenter might be a way to mitigate this possible bias.

Studies with simulations include one by Hulme and Bloebaum 2000, where they used the CASCADE software tool to create a representation of a multidisciplinary system. The authors themselves stated that this tool does not create a “real-world" optimization problem, but it is useful in that the multidisciplinary problems they simulated in CASCADE had very difficult design spaces being both non-convex with many local minima. They found that the MDF architecture required fewer iterations and found better optima than IDF. However, they noted that in “real-world" applications, IDF, while requiring more iterations, would converge in less time than MDF, albeit at a worse optimum. Recently, similar work was done by Zhang et. al. 2017, where differing MDAO architectures were tested as problem complexity varied. In this paper the number of disciplines and variables were controlled mathematically, and they found that MDF took less time than IDF to find an optimum in all cases, with the difference between the two increasing somewhat as complexity increased.

The studies conducted using more “real-world" problems include a similar study to this research done by Ajmera et al. 2004. In this paper MDAO architectures were analyzed using an aircraft model with aerodynamics and structural mechanics as two of the disciplines. This paper tested 6 different architectures, four variants of MDF and two variants of IDF. The optimization problem in that paper was to minimize the structural weight of the wing. Specifically, they minimized the components of the wing that bear an aerodynamic load. By adding in trim, the aircraft’s angle of attack,as a separate discipline, the authors were able to create their different MDF architectures by changing how and when the trim was either an intermediate variable or a design variable. They found that their optimizer was quite consistent between the four MDF architectures and the first

IDF architecture. The first IDF architecture required many more function calls and a considerably greater amount of computing time. These results used the equivalent plate method for structural analysis rather than the FEA used in the present study. This made it difficult to compare the results directly, as the equivalent plate method can be solved in less time than FEA.

Clearly, when optimizing any system, design teams want to achieve the optimal solution or solutions while limiting the time and computing resources required to get results. Therefore, evaluation of MDAO architectures frequently use number of function evaluations, optimality, and computing time as the most important metrics. Other characteristics in choosing an MDAO architecture include the previously mentioned managerial considerations referred to by Balling and Sobieski 1996. Perez et al. 2004 extended these metrics to include transparency, simplicity, and portability, finding that MDF is preferable to IDF in these, with the exception of portability.

C.3. METHODOLOGY

This section describes the development of the aircraft model case study and the setup of the MDAO architectures.

C.3.1. Model Development

The case study used for this research is a basic fixed-wing aircraft with four modeled disciplines: geometry, aerodynamics, structures, and performance. The geometry is modeled using a parametric modeling tool, the aerodynamics and structures are modeled with simulations, while the performance is modeled with analytic equations. Range is used for the objective of all optimization routines, using the Breguet equation [23].

The baseline design can be seen in Figure 1. Several assumptions are made in this aircraft model. First, the geometry and flight condition inputs are within a range to keep the model looking like a “normal" aircraft in stable flight conditions. Since the flight conditions are stable, the flow conditions for each wing are assumed to be symmetrical, meaning that structural analysis is performed on only one wing. The structure underlying the wing has been modeled as a simple cantilever beam, using the properties of Al 6061-T6, a common material for aircraft structures, for which the maximum stress constraint has been set to 1000 psi. The maximum fatigue strength of Al 6061-T6 is 14000 psi, so the stress constraint is significantly lower. However, when considering that stress will be much higher during maneuvers and a safety factor will be included, the maximum stress constraint is reasonable. The geometrical components maintain their general position relative to one another, and none can be added or taken away.

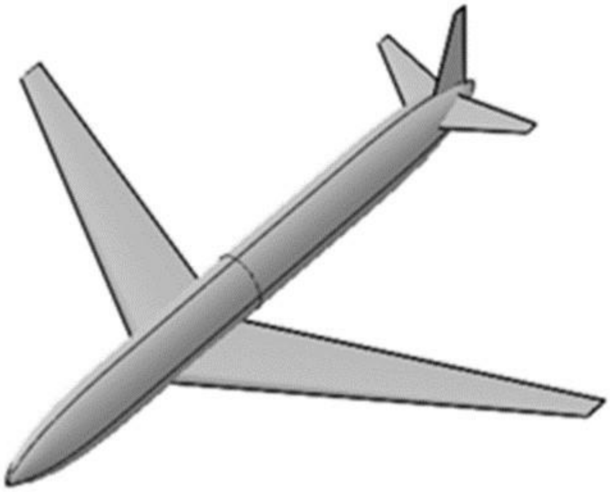


Fig. 1 Baseline Aircraft.

The geometry of the aircraft was created in NASA's OpenVSP parametric modeling tool [24]. OpenVSP has a selection of aircraft components such as wings and fuselages that designers can use. This preset library of components meshes the geometry in a format that can be used by OpenVSP's sister CFD tool, VSPAero [24], which was used for the aerodynamic analysis. While it is a separate program, it can be run from the OpenVSP GUI. VSPAero inputs the meshed geometry from OpenVSP and also requires inputs for the flight conditions. VSPAero is capable of running either the Vortex Lattice Method (VLM) or Panel Method for CFD. In this paper the VLM was used exclusively, largely because it has accurate results and a faster run time; furthermore, it is also less prone to failure in the VSPAero environment than the panel method. Since VSPAero requires the meshed geometry from OpenVSP, these two disciplinary models must be run sequentially in order to have aerodynamic outputs. This has some implications for possible architectures that can be constructed with this aircraft model. For instance, as will be seen later, the IDF architecture used in this study does not have the geometry and aerodynamics disciplines independent of one another.

With inputs from the geometry component, the aerodynamics solver outputs the overall coefficients of lift and drag, respectively C_l and C_d, which are used for the performance component that finds the range of the aircraft. This solver also outputs the coefficients of force, C_{f_i}, for every span-wise cross-sectional element. In this paper, there are 15 cross-sectional elements on the wing. The coefficients of force are used in the structures component to find the loads on

the wing. These loads are used with an FEA MATLAB code [25, 26]. Like the mesh for CFD, there are 15 elements representing the beam for each wing; because the first node is fixed and the last node is free, the stress is calculated at 14 nodes.

These disciplinary models take different amounts of time to run, which has an effect on the results found in this study. The performance and structures components take less than one second to run, the geometry component takes around five seconds, and the aerodynamics component takes between one and two minutes. These times represent the model being run on a two core 2.20 GHz processor with four GB of RAM. Clearly, to minimize the duration of finding optima using these disciplinary models, the CFD simulation underlying the aerodynamics component should be run as few times as possible.

The disciplinary models in this MDAO workflow were brought together and analyzed using the ModelCenter software package. ModelCenter wraps analysis tools so that different disciplinary models using different tools can more easily interface with one another. It also has built-in optimization algorithms, data visualization and analysis tools. ModelCenter aids in setting up MDAO architectures and provides an MDAO framework that can help to remove biases in the results which might arise from a designer better understanding how to set up one architecture [1]. In this framework OpenVSP and VSPAero are run in batch mode by using Microsoft Windows command line arguments. The MATLAB FEA code is run with a built-in ModelCenter plugin, and the analytic performance equations are done with an Angelscript code, also built into a ModelCenter scripting component.

Both of the optimization architectures require 11 design variables (inputs), these are as follows: Mach number (airspeed) and angle of attack for flight conditions, wing x-relative location (location along the fuselage), wing rotation angle, horizontal stabilizer rotation angle, and the span and chord for each of the wing, horizontal stabilizer, and vertical stabilizer to define the aircraft geometry. The input variables and their minima, maxima, and initial values are provided in Table 1. The initial values represent the original design of this model.

Table 1 Aircraft Geometry and Flight Conditions Input Values.

Parameter	Min	Max	Initial
Wing Span (ft)	20	50	25
Wing Chord (ft)	2	6	4
Wing Rotation (deg)	0.1	4	1
Wing X Location (ft)	8	12	9
HStab Span (ft)	6	11	8
HStab Chord (ft)	1	3	1.5
HStab Rotation (deg)	-5	-0.1	-1
VStab Span (ft)	6	11	8
VStab Chord (ft)	1	3	1.5
Mach #	0.2	0.35	0.3
Angle of Attack (deg)	0.1	5	1

HStab = horizontal stabilizer, VStab = vertical stabilizer

As mentioned before, the optimization algorithm has an effect on the computer resources required and optimality of the final solution. Both of the architectures were tested using the ModelCenter proprietary “Design Explorer" algorithm. This works by first running an Orthogonal Array Design of Experiments (DOE). The number of iterations depends on the number of global input variables the model has, which, for these analyses, required between 98 for MDF and 128 for IDF, due to the added coupling variables. A DOE generates a sample of the design space by evaluating the model at several levels along the entire range of each input variable.

From this sample, ModelCenter creates surrogate models using interpolating Kriging models and optimizes these models by using the Sequential Quadratic Programming (SQP) algorithm. This gradient-based algorithm is run several times from different random starting points. The stochastic nature of choosing these starting points leads to variability seen in the final results. The optima found in this step are then run using the actual model; these results are used to further refine the surrogates which are again optimized using SQP. Finally, a local pattern search tries to find an improved design near the current best design. If one is found, then the process iterates after refining the surrogates again. If not, then the optimal solution has been found.

The Design Explorer algorithm is used for this study mainly because of its robustness when the model has a failed run. It also can find a solution in less time than a genetic algorithm. Also, due to its use of surrogate models, it takes a shorter time than other algorithms that can handle failed runs, such as genetic algorithms, which were run on this model twice but were stopped after 40 hours without converging.

C.3.2. MDF Architecture Setup

This paper compares the MDF and IDF architectures. In the case of the MDF architecture, the ModelCenter workflow was set up as a process where all of the disciplinary models were run in the following order: geometry, aerodynamics, performance, structures.

The MDF architecture was set up in ModelCenter using a “process" workflow. The process workflow in ModelCenter runs all of the disciplinary models sequentially, and none of the models can be run independently unless separate loops are set up. Figure 2 shows a visual representation of this workflow. For simplicity, this workflow does not illustrate the data flow within this MDAO problem, focusing only on the steps taken for each algorithm iteration.

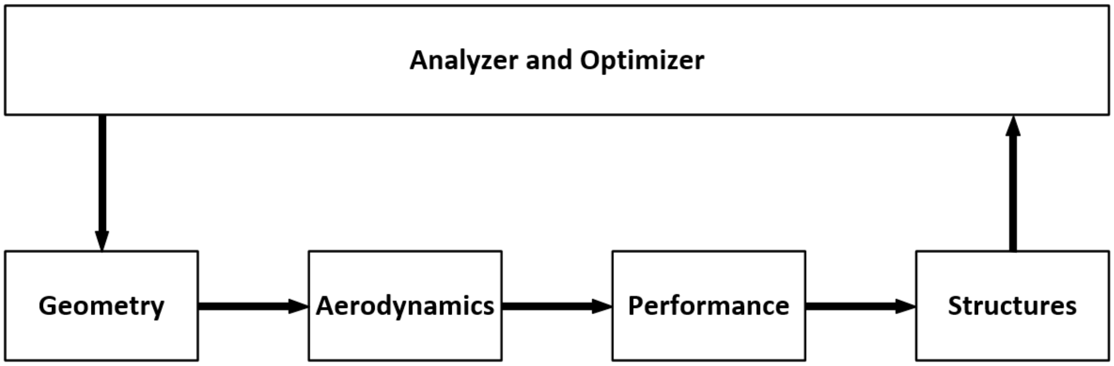


Fig. 2 MDF Architecture Workflow.

C.3.3. IDF Architecture Setup

For the IDF architecture the ModelCenter “data" workflow style was used; this allows disciplinary models to be run independently. As mentioned in the background section, coupling variables are required in order to ensure that the inputs and outputs between disciplinary models are consistent. The important coupling variables for this architecture were chosen to be the coefficients of force along the span-wise direction of the wing, which are outputs of the CFD model and inputs to the FEA model.

As stated previously, there are 14 nodes where aerodynamic loads were calculated, which would create 14 separate coupling constraints. In order to avoid the “curse of dimensionality," and recognizing the coupled nature of those 14 values, this 14x1 vector was reduced to a 2x1 vector using the reduced representation method of Proper Orthogonal Decomposition (POD). POD is a way to reduce the number of coupling constraints while still maintaining an accurate representation of the data [27]. In practice, this was done by adding another component in ModelCenter that takes theforce coefficient vector and performs POD. Then, the structures component, having been decoupled from aerodynamics, adds two input variables for the POD coefficients. Having fewer coupling constraints by reducing the coefficient of force vector will also reduce the number of inputs in the FEA component that need to match the outputs of the CFD; this aids convergence of the optimization algorithm since more design points are multidisciplinary feasible. Even after such a large reduction, the reconstructed Cf vector had a cumulative percentage variance of higher than 99.98 percent. In the actual optimization formulation, these coupling variables were not set to be exactly equal to one another but to be within a small range; these ranges have been relaxed from the 0.00001 consistent with other studies [18]. The first POD coefficient has a range of -1.245 to 1.5218 and is constrained to within 0.01. The second POD coefficient has a range of

-0.0865 to 0.0383 and is constrained to within 0.001.

Since creating the POD coefficients requires a sample of the data, a 500-run Latin Hypercube DOE, similar to the Orthogonal Array used by the Design Explorer algorithm, was initially run using the MDF architecture.

In the IDF architecture, the Mach number is set with the other design variables. This speeds up the analysis compared to if it were set with VSPAero, since in ModelCenter when this input is set, the rest of the components needing this input will be run. This would require running the CFD component every time, but it is not necessary.

The IDF architecture workflow can be seen in Figure 3. The major differences in Figure 3 compared to Figure 2 is the decoupled structures disciplinary model; also, the coupling constraint data flow is shown.

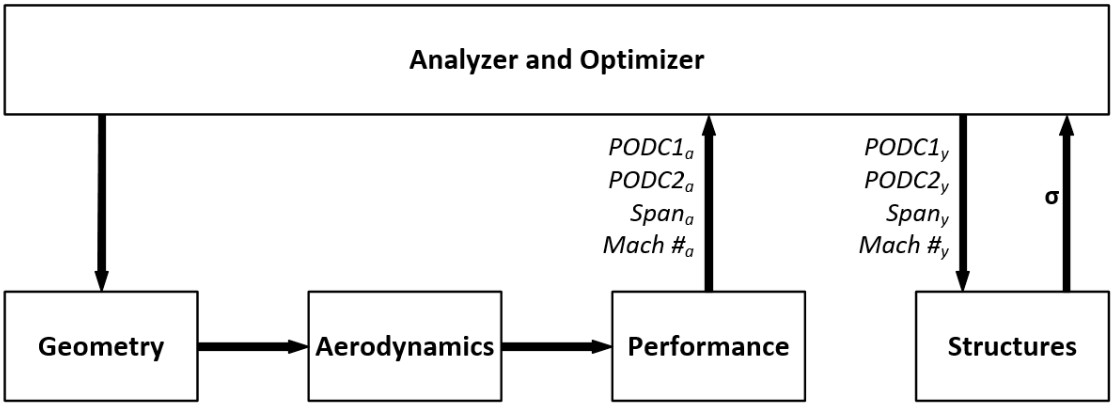


Fig. 3 IDF Architecture Workflow.

$PODC1$ and $PODC2$ are the POD coefficients, $Span$ is the wing span, $Mach\ \#$ is the airspeed, and $\boldsymbol{\sigma}$ is the vector of stresses which used the coefficient of force vector recreated by reversing the POD process. The subscripts a and y denote respectively whether the variable is an analysis tool output or coupling constraint input, following the conventions of Equation (2).

C.4. RESULTS

Ten optimization routines were run for each architecture using the Design Explorer algorithm. Each of the twenty total runs were identical, the differences in optimum and computing time are due to the stochastic nature of the Design Explorer algorithm, as described in the previous section. The results of these routines for the optimality of range, the total time elapsed, and the time per run can be seen in Table 2. The best results among the ten routines for each architecture and across the averages for both architectures are highlighted in green; the worst among the ten routines are highlighted in red.

On average, optimization using the MDF architecture had slightly higher than a 3.5 percent increase in optimality over the IDF architecture. The best optimum for the MDF architecture was significantly better than that for the IDF, with more than an 11 percent improvement; the worst optima were closer, with the MDF having a higher range than the IDF worst solution by 5.7 percent. This validates and clarifies the benefit in optimality of MDF over IDF.

While MDF offers a small increase in optimality, this is gained at a significant 63 percent increase in total time elapsed. While the three routines with the shortest time elapsed were both MDF, the five routines that took the longest time were also MDF, the longest two of which took nearly 30 hours, more than double the amount of time of the slowest IDF routine. Similar to time elapsed, the IDF architecture also had a shorter time per run, by 26 percent, although the best and worst values did not deviate far from the mean.

	MDF			IDF		
Routine #	Range (mi)	Time elapsed (hr)	Time per run (min/run)	Range (mi)	Time elapsed (hr)	Time per run (min/run)
1	9674.1	21.62	1.97	**9653.0**	7.47	1.57
2	9124.3	7.56	1.69	9039.4	8.52	**1.69**
3	9416.6	5.03	**1.66**	8814.7	7.06	1.43
4	**8908.4**	**4.63**	1.81	9185.7	**13.96**	1.62
5	9412.3	21.56	1.92	8871.5	7.50	1.59
6	9115.7	17.11	1.97	9421.9	10.74	1.34
7	9117.1	6.93	1.98	9527.0	7.19	**1.22**
8	9711.1	9.25	1.96	9158.1	11.56	1.45
9	9134.0	29.09	1.98	8942.8	**6.98**	1.48
10	**10724.5**	**29.44**	**1.98**	**8426.1**	12.58	1.64
Average	**9433.8**	15.22	1.89	9104.0	**9.36**	**1.50**
p-values	0.06	0.05	0.00			

Table 2 Results by Architecture

A student's T-test was performed in order to capture the significance of these preliminary results. The p-values from this test can be seen in the lower left portion of Table 2. As expected, there is a very high confidence level that the time per run in IDF is lower than the time per run in MDF. The range and time elapsed have confidence levels of 94 and 95 percent, respectively. This calls for further routines to be run in order to better characterize the results, especially for range.

Overall, these results confirm that the MDF architecture can find a better result while taking longer than IDF. The optimality results match the results found by Hulme and Bloebaum 2000, and, if this improvement were to hold up over additional runs to improve the statistical confidence, then these results would fit with their predictions, which were that MDF solutions would take longer when applied to real-world MDAO problems.

C.5. CONCLUSIONS AND FUTURE WORK

This paper details a comparison study of the MDF and IDF MDAO architectures. The study extends previous work by using simulations to model the system being optimized, a fixed-wing aircraft. The results confirm that the MDF architecture is preferable when it comes to finding the optimal design, yet it requires more computing resources and a more coordinated model integration effort to do so.

Future directions for this research are to use this model within the ModelCenter framework to benchmark other MDAO architectures. Using different algorithms and seeing how they affect the optimality of designs and computing resources can also contribute to our understanding of MDAO architectures.

Further work on developing the model will also be done, including extending the FEA to include the wing skin and a more representative underlying wing structure. Other extensions to the model include adding mission profiles and maneuvers that may cause more stress on the structure. Finally, including more disciplinary models such as a payload or communications architecture is planned. Extending the model along with using a finer CFD mesh will increase the computing resources required for optimization runs and should help to evaluate MDAO architectures in circumstances that would be similar to how they are used in defense and industry applications.

C.6. References

[1] Martins, J. R., and Lambe, A. B., "Multidisciplinary design optimization: a survey of architectures," *AIAA journal*, Vol. 51, No. 9, 2013, pp. 2049–2075.[2] Sobieszczanski-Sobieski, J., "Sensitivity analysis and multidisciplinary optimization for aircraftdesign-Recent advances and results," *Journal of Aircraft*, Vol. 27, No. 12, 1990, pp. 993–1001.

[3] Balling, R. J., and Sobieszczanski-Sobieski, J., "Optimization of coupled systems-a critical overview of approaches," *AIAA journal*, Vol. 34, No. 1, 1996, pp. 6–17.

[4] Phoenix Integration, "Phoenix Integration Homepage," online, March 2018. URL `https://www.phoenix-int.com/`.

[5] Schmit, L. A., "Structural design by systematic synthesis," *Proc. of Second Conference on Electronic Computation ASCE, New York*, 1960, pp. 105–122.

[6] Braun, R. D., Moore, A. A., and Kroo, I. M., "Collaborative approach to launch vehicle design," *Journal of spacecraft and rockets*, Vol. 34, No. 4, 1997, pp. 478–486.

[7] Balling, R., and Rawlings, M., "Collaborative optimization with disciplinary conceptual design," *Structural and multidisciplinary optimization*, Vol. 20, No. 3, 2000, pp. 232–241.

[8] He, Y., and Mcphee, J., "Multidisciplinary optimization of multibody systems with application to the design of rail vehicles," *Multibody System Dynamics*, Vol. 14, No. 2, 2005, pp. 111–135.

[9] Grossman, B., Gurdal, Z., Haftka, R., Strauch, G., and Eppard, W., "Integrated aerodynamic/structural design of a sailplane wing," *Journal of Aircraft*, Vol. 25, No. 9, 1988, pp. 855–860.

[10] Haftka, R. T., "Automated procedure for design of wing structures to satisfy strength and flutter requirements," 1973.

[11] Livne, E., Schmit, L., and Friedmann, P., "Towards integrated multidisciplinary synthesis of actively controlled fiber composite wings," *Journal of Aircraft*, Vol. 27, No. 12, 1990, pp. 979–992.

[12] Sobieszczanski-Sobieski, J., "A linear decomposition method for large optimization problems. Blueprint for development," 1982.

[13] Lasdon, L. S., *Optimization theory for large systems*, Courier Corporation, 1970.

[14] Cramer, E. J., Dennis, J. E., Jr, Frank, P. D., Lewis, R. M., and Shubin, G. R., "Problem formulation for multidisciplinary optimization," *SIAM Journal on Optimization*, Vol. 4, No. 4, 1994, pp. 754–776.

[15] Papalambros, P. Y., and Wilde, D. J., *Principles of optimal design: modeling and computation*, Cambridge University Press, 2000.

[16] Tedford, N. P., and Martins, J. R., "Benchmarking multidisciplinary design optimization algorithms," *Optimization and Engineering*, Vol. 11, No. 1, 2010, pp. 159–183.

[17] Haftka, R. T., Sobieszczanski-Sobieski, J., and Padula, S. L., "On options for interdisciplinary analysis and design optimization," *Structural optimization*, Vol. 4, No. 2, 1992, pp. 65–74.

[18] Kodiyalam, S., "Evaluation of Methods for Multidisciplinary Design Optimization (MDO). Phase 1," 1998.

[19] Hulme, K., and Bloebaum, C., "A simulation-based comparison of multidisciplinary design optimization solution strategies using CASCADE," *Structural and Multidisciplinary Optimization*, Vol. 19, No. 1, 2000, pp. 17–35. doi:10.1007/s001580050083, URL `https://doi.org/10.1007/s001580050083`.

[20] Zhang, D., Song, B., Wang, P., and He, Y., "Performance Evaluation of MDO Architectures within a Variable Complexity Problem," *Mathematical Problems in Engineering*, Vol. 2017, 2017.

[21] Ajmera, H., Mujumdar, P., and Sudhakar, K., "MDO architectures for coupled aerodynamic and structural optimization of a flexible wing," *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, 2004, p. 1543.

[22] Perez, R., Liu, H., and Behdinan, K., "Evaluation of multidisciplinary optimization approaches for aircraft conceptual design," *10th AIAA/ISSMO multidisciplinary analysis and optimization conference*, 2004, p. 4537.

[23] Shevell, R., "Fundamentals of Flight. 2nd," , 1989.

[24] OpenVSP, "OpenVSP Homepage," , March 2018. URL `http://openvsp.org/`.

[25] Tawfik, M., "FEM: Beam FreeMat (Matlab) Code," [online], Accessed April 15, 2018, April 2015. URL `https://www.youtube.com/watch?v=tgBLjHHAJY0`.[26] MATLAB, *version 8.6.0 (R2015b)*, The MathWorks Inc., Natick, Massachusetts, 2015.

[27] Alexander, M., Allison, J., and Papalambros, P., "Decomposition-based design optimisation of electric vehicle powertrains using proper orthogonal decomposition," *International Journal of Powertrains*, Vol. 1, No. 1, 2011, pp. 72–92.

D. MDAO AND GRAPHICAL CONOPS

Authors: Brian Chell, Steven Hoffenson, Douglas Ray, Roger D. Jones, Mark R. Blackburn

Optimizing for Mission Success in Highly Stochastic Scenarios

Stevens Institute of Technology, Hoboken, NJ 07030

Optimization under uncertainty increases the complexity of a problem as well as the computing resources required to solve it. With higher levels of uncertainty, these difficulties are exacerbated. However, when optimizing for mission-level objectives, rather than component- or system-level objectives, an increase in uncertainty is inevitable. Previous research has found methods to perform optimization under uncertainty, such as robust design optimization or reliability-based design optimization, but this is generally done with problems like component tolerances that do not capture the high amounts of stochasticity of a mission-level problem. In this paper, an approach for formulating and solving highly stochastic mission-level optimization problems is described. A case study is shown using an unmanned aerial system (UAS) on a search mission while an "enemy" UAS attempts to interfere. This simulation, modeled in the Unity Game Engine, has highly stochastic outputs, where the time to mission success varies by multiple orders of magnitude, but the ultimate goal is a binary output representing mission success or failure. The results demonstrate the capabilities and challenges of optimization in these types of mission scenarios.

D.1. Introduction

Optimization under uncertainty is a difficult problem to solve, yet it is essential given the many different sources of uncertainty in real life situations. While robust design optimization (RDO) and reliability-based design optimization (RBDO) have been successfully used to optimize designs in situations where the uncertainty is low, such as in component tolerances, their applicability to scenarios with very high levels of uncertainty is limited. Both methods assume that the problems have well-defined constraints and well-known levels of uncertainty that can be analytically modeled. However, when optimizing for mission success, where constraints are poorly-defined or unknown, many sources of epistemic and aleatory uncertainty are present, and unexpected or emergent behavior may occur, alternative optimization approaches are needed.

In this paper, alternative approaches to optimizing under uncertainty are proposed and examined using a case study that involves a highly stochastic mission-level scenario. In this reconnaissance scenario, which is described in detail in Section IV, one unmanned aerial system (UAS) is searching for a target while a counter-UAS tries to interfere. The simulation has many input variables related to the design of the UAS and produces highly stochastic results, raising a number of challenges for optimization. One of these challenges for this and any scenario with high uncertainty is that many thousands of simulations will need to be run in order to capture the likelihood of mission success across the design space. Running this many simulations may require a significant amount of time as well as extensive computing resources. Another challenge is that there are currently only two outputs that can be used for optimization: a continuous parameter representing the time to find the target and a binary parameter representing mission success or failure.

By examining and attempting to optimize the design of the UAS in this simulation, several methods will be formulated and tested to address this type of highly-stochastic problem. The purpose of these studies is to find a generalizable approach for these types of problems that can provide meaningful results while limiting the amount of time and computing resources required to solve them.

D.2. Background

This section presents a short review of the relevant literature, discussing previous work in optimization under uncertainty, simulation optimization, and mission-level optimization.

D.2.1. Optimization under Uncertainty

In scenarios with high levels of uncertainty, deterministic optimization of mathematical or simulation-based models is often inadequate. This is because inherent uncertainties in manufacturing or environmental parameters will propagate into uncertainties in the optimization objective, making it suboptimal or causing it to not meet constraints [1]. In order to overcome these pitfalls, these uncertainties need to be accounted for. In the first case, using robust design optimization (RDO) is a possible solution. Innovated by G. Taguchi [2], RDO seeks to find designs that are more "robust," which, in this case means that the objective is less sensitive to uncertainty in the input parameters, maintaining a more optimal solution over the range of uncertainty [3]. On the other hand, the case where uncertainty causes the optimization problem to not satisfy constraints, reliability-based design optimization (RBDO) is a suitable method. Here, the optimization constraints add a term representing the allowable probability, chosen by the designers, of the constraint being violated. Both RDO and RBDO model uncertainty in the inputs and constraints.

Stochastic programming and fuzzy programming are also fundamental methods for optimization under uncertainty [4]. However, stochastic programming utilizes programming with recourse, and it is not appropriate for design scenarios where all design decisions must be made up front, such as in the case study of the present paper. Fuzzy programming uses imprecisely-defined values for inputs and constraints, and it is best-suited to problems where there are different ways to subjectively interpret the optimization problem formulation, which is also not appropriate for the present mission-level scenario.

D.2.2. Simulation Optimization

As system complexity increases, it becomes difficult or sometimes computationally intractable to optimize with only analytic equations [5]. Using a simulation to model and optimize these complex systems is a common solution. Optimizing a computer simulation has the structure as any optimization problem defined by analytic equations, which is to maximize or minimize an objective or objectives while meeting a set of constraints. With computationally expensive simulations, it is especially important to make the search for the best possible set of inputs as efficient as possible [6].

Following the lead of Barton and Meckesheimer [7], the choices available for simulation optimization can be broken down into four categories: random search and metaheuristics, ranking and selection, direct gradient methods, and surrogate model methods. Figure 1, adapted from their book chapter, can help users decide which of these strategies to implement. In the case study used in this project, the input vector, $\mathbf{x}$, could be treated as either a discrete set or as continuous. This is due to the fact that the simulation only allows integer inputs, and one of the decision variables has a range of 1 to 3. However, other decision variables have large ranges and can be thought of as continuous. Therefore, random search and metaheuristics are applicable to this case. The objective, f, is a continuous value, yet it is unknown whether it is differentiable, which suggests that surrogate model methods would also be applicable. Therefore, this subsection of the background focuses on these two methods.

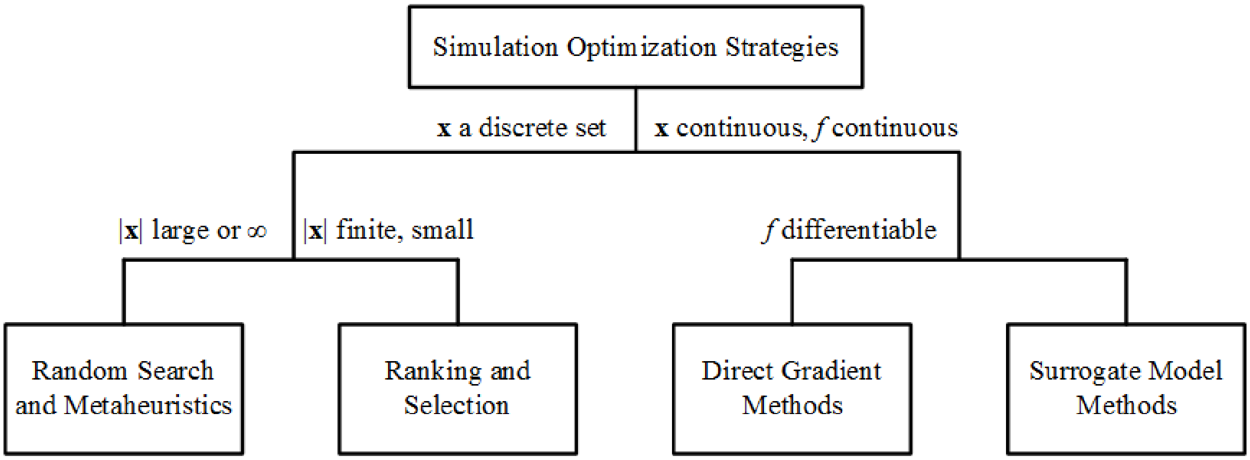


Fig. 1 Simulation Optimization Strategies Flowchart (adapted from [7])

Often, when working with simulations not much, if anything, is known about the underlying equations being used; only the inputs and outputs are able to be analyzed. In this case the simulation is known as a "black box". Black box optimization is a sub-discipline of simulation optimization. Since the underlying functions and the existence or nonexistence of their gradients are unknown, different methods must be used than when working with a purely analytic optimization problem.

D.2.2.1. Random Search and Metaheuristics

Random search methods select a sample of of initial design points and then run the simulation to evaluate them. Using the results from these evaluations, the optimal solution is estimated and then a new sampling strategy is selected, based on the "neighborhood" of where the optimal solution is expected to be. The updated sampling strategies can use simple methods like a descent algorithm to find optimal points or more complicated and efficient methods. [8]. Selecting more complicated strategies for sampling is where random search methods blur with metaheuristics.

Metaheuristic techniques are similar to random search in that an initial set of solutions is found, and then the design space is explored for improvement. Their benefit over basic random search methods is that this improvement is done in a more efficient manner. These methods include simulated annealing, tabu search, and genetic algorithms [9].

D.2.2.2. Surrogate Model Methods

Another way to optimize a simulation is by sampling the design space and creating a surrogate model. Doing this will not only enable the use of gradient-based optimization algorithms, but also will reduce the amount of time required for each run of the optimization routine. It should be noted that while each optimization run will be shorter when using a surrogate model; the overall time for the optimization routine might take longer due to the time required for proper sampling and model creation [10].

Response surface approximation is the most common method of creating these surrogate models. This general approach takes the output from the sample and uses some response surface methodology to create a predictive model of the simulation. First or second-order polynomial regression models are commonly used, and spline regression can be used if higher-order polynomial regression is needed to create a better fit [7].

Another surrogate modeling method is kriging. Developed by Danie G. Krige [11] to help find gold in South Africa, kriging predicts the value of the optimization objective by combining a global polynomial model, similar to that found in a response surface, and a term representing the systematic departure in order to infer local effects [12, 13]. Kriging generally creates smoother surrogates that are less sensitive to local effects [7].

D.2.3. Mission-Level Optimization

Mission-level optimization is a term that can take several different meanings. There are largely two seen in the literature. The first is from optimizing autonomous robots to have a better success/failure ratio when performing tasks such as maneuvering over an obstacle [14] or grasping an object [15]. These studies are similar to this one in that the key performance indicator (KPI) is a binary success or failure output; however, they use machine learning techniques in order to change the robots' behavior in order to maximize success. For the case study in this project, the hardware designer has no control over the search strategy of the UAS.

The second is seen in optimizing aerospace vehicles, especially space systems. This is mostly a semantic distinction, because aerospace engineers often speak of the systems they design as performing missions, even though the KPIs in these cases are continuous variables being optimized with methods that could fit more general problems[16–19]. There are similarities in these mission-level optimization problems that take into account several phases, such as takeoff, cruise, and landing, while this case study has phases of search and avoid. The key difference that makes this study unique is the focus on the binary success/failure output.

D.3. Case Study

The studies in this paper have been conducted on a mission simulation built in the Unity Game Engine. In this simulation, a blue UAS is searching for a target in a suburban environment. The blue UAS is modeled with physics-based equations and has a limited battery capacity, and it follows a random flight path while searching for the target. If the blue UAS finds the target before crashing due to depleted batteries, the mission is considered to be a success. Increasing the complexity and uncertainty of the scenario, there is also a red "enemy" UAS that is maneuvering itself autonomously to block the path of the blue UAS. This causes the blue mission to fail more often, as it is using energy trying to avoid the red UAS and may also miss the target due to the interference in its field of view. Figure 2 shows a snapshot of the simulation in progress.

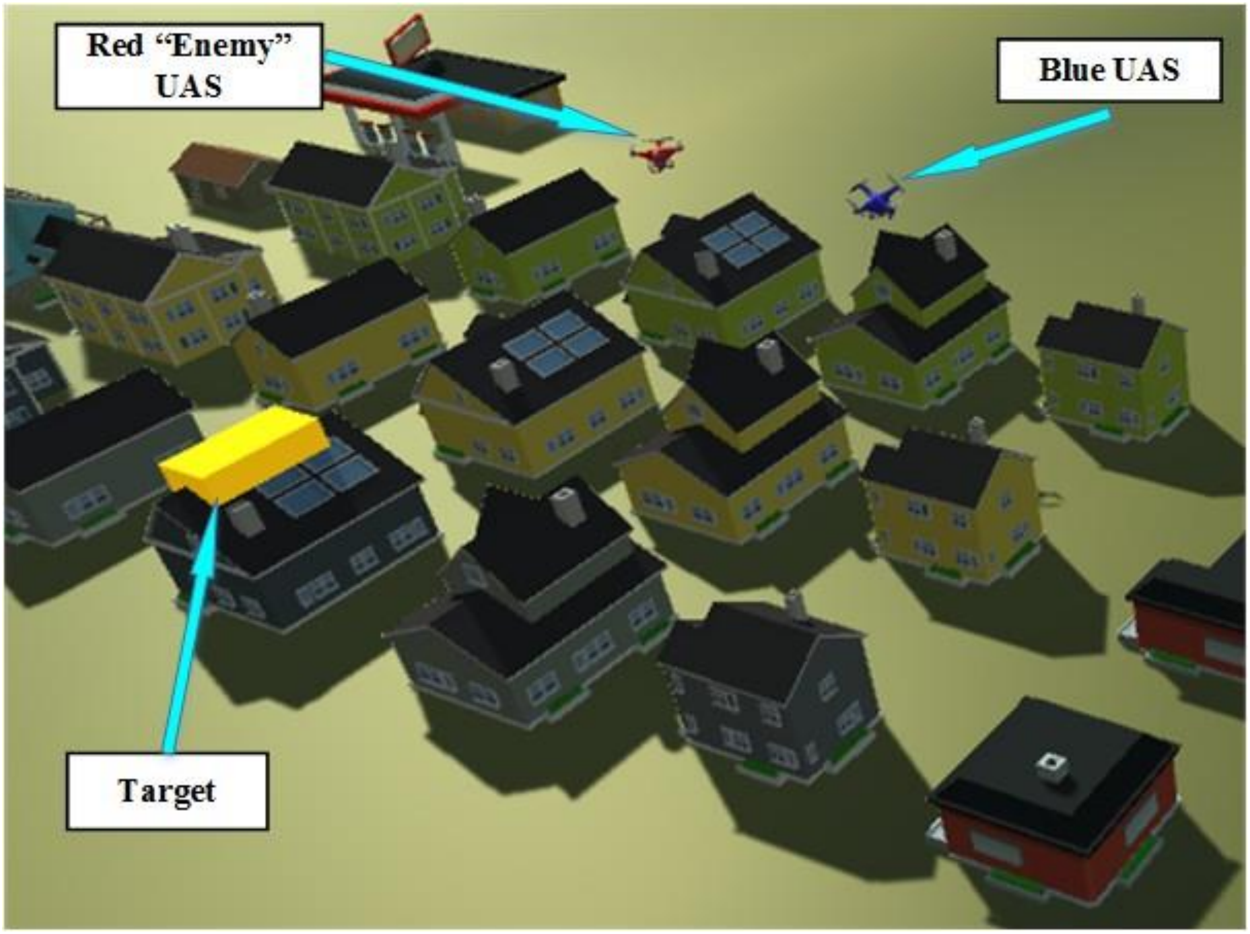


Fig. 2 Snapshot of UAS/Counter-UAS Scenario

This simulation has many inputs that can affect mission success or failure. Each UAS is defined by nine different design variables that characterize its geometry and power characteristics, including battery charge, battery voltage, battery specific energy, battery life, battery canister mass, UAS frame mass, a spring constant that controls how far the UAS will overshoot when it turns, rotor radius, and UAS total volume. Several of these design variables are lightly constrained, and they are allowed to vary by more than an order of magnitude. However, even when the design variables are held constant, individual runs of the simulation show high amounts of stochasticity. One set of simulations found a range in the time that the blue UAS took to find the target from a low of 1 second to a high of 25 minutes.

While there is a high amount of complexity both with the inputs to the model as well as within the model itself, the KPI in this example is a binary output representing mission success or failure. The question of how to optimize such a complex system with this very basic and stochastic output is unique to this research and the key research question that this study sets out to address.

D.4. Methodology and Initial Results

In order to simplify analysis, the Unity simulation was wrapped in the Phoenix Integration ModelCenterTM multidisciplinary design analysis and optimization (MDAO) software package. This allows the model to be analyzed hundreds or even thousands of times efficiently, as these runs are executed “headless,” meaning that the Unity graphical user interface (GUI) is not shown nor is direct user input required.

Due to the high stochasticity in results even when the design variables are held constant, each run was repeated 20 times in order to better characterize the different designs. By repeating each design 20 times, the binary output of success/failure can be analyzed as a probability of crashing, $P(crash)$, for each design point; this allows it to be analyzed and optimized as a continuous variable instead of the binary success/failure. In order to efficiently explore the design space, a Design of Experiments (DOE) was run using the Definitive Screening Design (DSD) sampling method. The DSD assesses each input at three levels, which gives some sense of the general impact these inputs have on the objectives [20]. The DOE evaluated 26 design points, and so with each design repeated 20 times, a total of 520 simulations were executed. The percent of crashes per design point is seen in Fig. 3. Of the 26 designs evaluated, 18 of them had a 100 percent success rate. The designs that had failures show a wide range of success rates, with design 24 having zero successful runs.

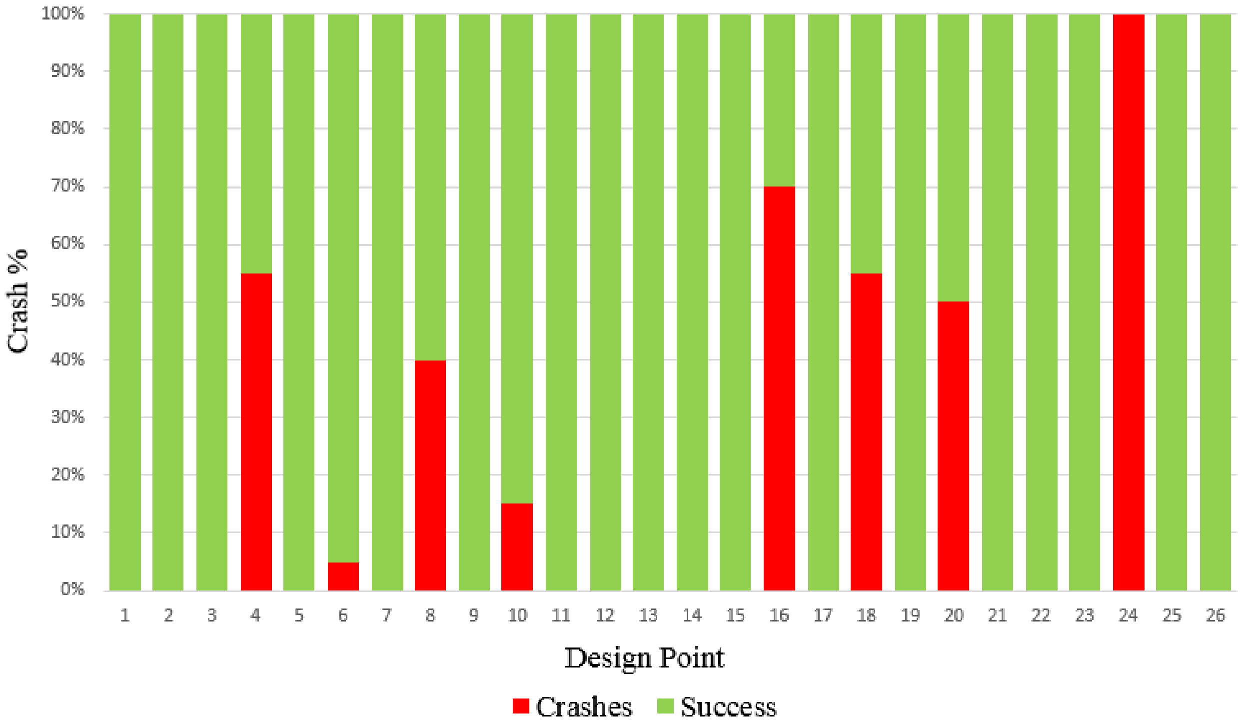


Fig. 3 Percent of Crashes for Each Design Point

While mission success is the KPI for this study, an important intermediate variable, the time to find the target, has also been used to find the best designs. By minimizing the mean time to find the target, $MTTF$, the mission success rate will also, generally, increase. However, this has unintended consequences, both negative and positive, that require further exploration. These unintended consequences arise from how the case study measures the time the UAS takes to find the target, regardless of whether the mission is a success or a failure. That means a design with a weak battery will actually have a low $MTTF$, because the battery life will put an upper limit on the amount of time the UAS takes to find the target regardless of mission success. Positive consequences from this feature are that if the optimization problem is formulated to minimize both $P(crash)$ and $MTTF$, it could result in Pareto set giving designers tradeoffs of designs that are both successful, and likely inexpensive, if battery constraints could be relaxed.

Figure 4, which includes *MTTF* as well as the standard deviation of the mean time, *StDTTF*, shows these unintended consequences. It shows that the designs with both the fastest times and most consistent results are numbers 16 and 24. As we know from Fig. 3, these are the designs that had the worst success rate, which actually makes them the worst designs for the purpose of this project.

Using the DOE data, surrogate models for the three objectives were created. For P(crash), a generalized linear model with binary logistic regression using penalized likelihood was used [21, 22]. The surrogates for MTTF and StDTTF were loglinear variance models fit for mean and variance [23]. These models were then used to create a utility function to find the best design. The utility coefficients used weights of 5 for *P(crash)*, 3 for *MTTF*, and 1 for *StDTTF*. These reflect that mission success/failure is our KPI. This surrogate model predicted that the optimization solution would have a *P(crash)* of less than 2 percent, *MTTF* 69.1 s, and *StDTTF* of 51.7 s. When this design was run 20 times there were zero failures, an *MTTF* of 55.4 s, and a *StDTTF* of 52.6 s, fitting the prediction very well. These values represent a significant upgrade over the other design points with a 100 percent success rate. This can be seen in Fig. 5 where the optimization solution is clearly better than any solution without crashes.

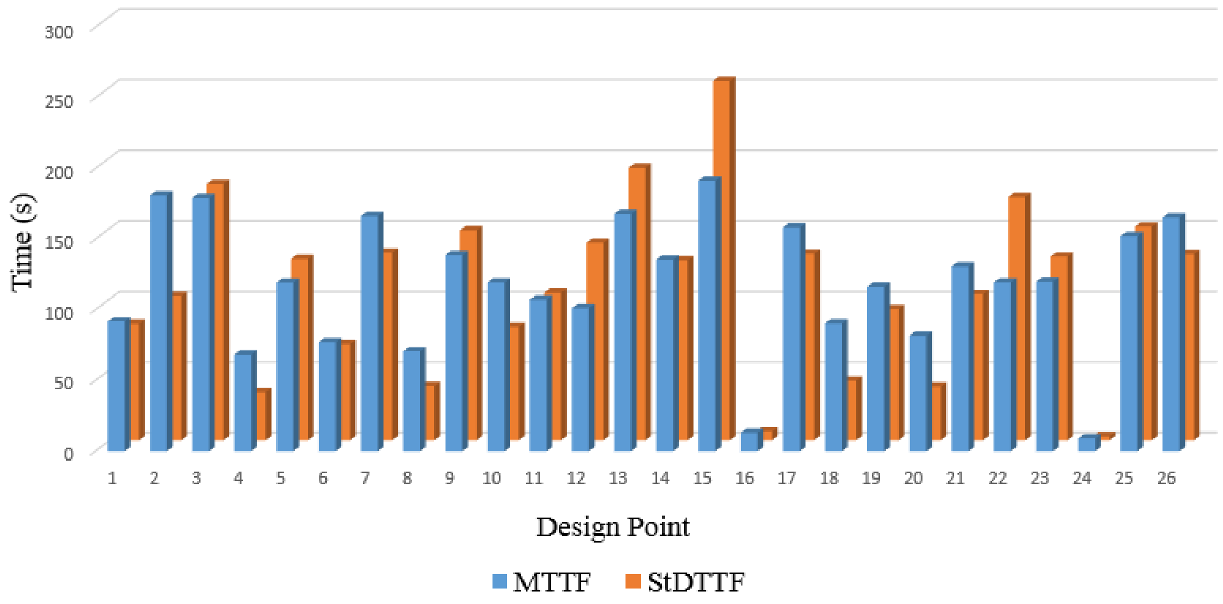


Fig. 4 Mean and Standard Deviation of Time to Find Target for Each Design Point

D.5. Next Steps

In order to find the best way to optimize a mission-level problem, some other methods should be tested on this model. Some planned next steps include using different DOE sampling strategies, different surrogate modeling strategies, and using the current surrogate and changing the weights used in the utility functions.

In order to further refine the surrogate model, further DOEs will be conducted. These DOEs will run the simulation more than 20 times at each design point to see how consistent the results are and how they compare to other design points, and also to achieve a reliable measure of the probability of mission success at each design. Sensitivity analyses will be performed during this process in an effort to reduce the dimensionality of the problem. Due to a lack of knowledge of the design space and a non-continuous objective function, evolutionary algorithms will be applied first; then, different algorithms will be tested to see their efficacy in finding solutions.

Another approach to test is to use random search and metaheuristics methods to optimize this model. This could be done by minimizing MTTF as a single-objective problem, or both MTTF and StDTTF as a multi-objective problem while including the probability of crashing as an inequality constraint which must be less than, for instance, 0.5%. This is a similar approach to that used in the aerospace mission-level optimization problems except in those cases the probability of failure was 0%.

These methods will be compared and contrasted by seeing how many simulation runs are required to evaluate the computing resources required. This will be straightforward; however, comparing the solution optimality of each method will be more difficult as they will not all have some of the objectives. In order to compare methods for optimality, minimizing $P(crash)$ will be emphasized as the goal is optimization at the mission level. Other metrics, such as ease and time required for implementation will be studied while understanding that these will have more subjective measures.

D.6. Conclusions

Through the process of investigating this simulation and discovering which approaches work best in its optimization, this study brings us closer to a generalizable method of formulating and solving optimization problems involving highly stochastic mission scenarios. In addition, this work sheds light on the challenges and unique properties of these types of optimization problems, which can lead to strategies that may help to overcome specific difficulties in optimizing mission-level scenarios. Finally, this work advances the discussion and state-of-the-art regarding how to efficiently and effectively handle uncertainty in design, and it may lead to new avenues for further research in the field.

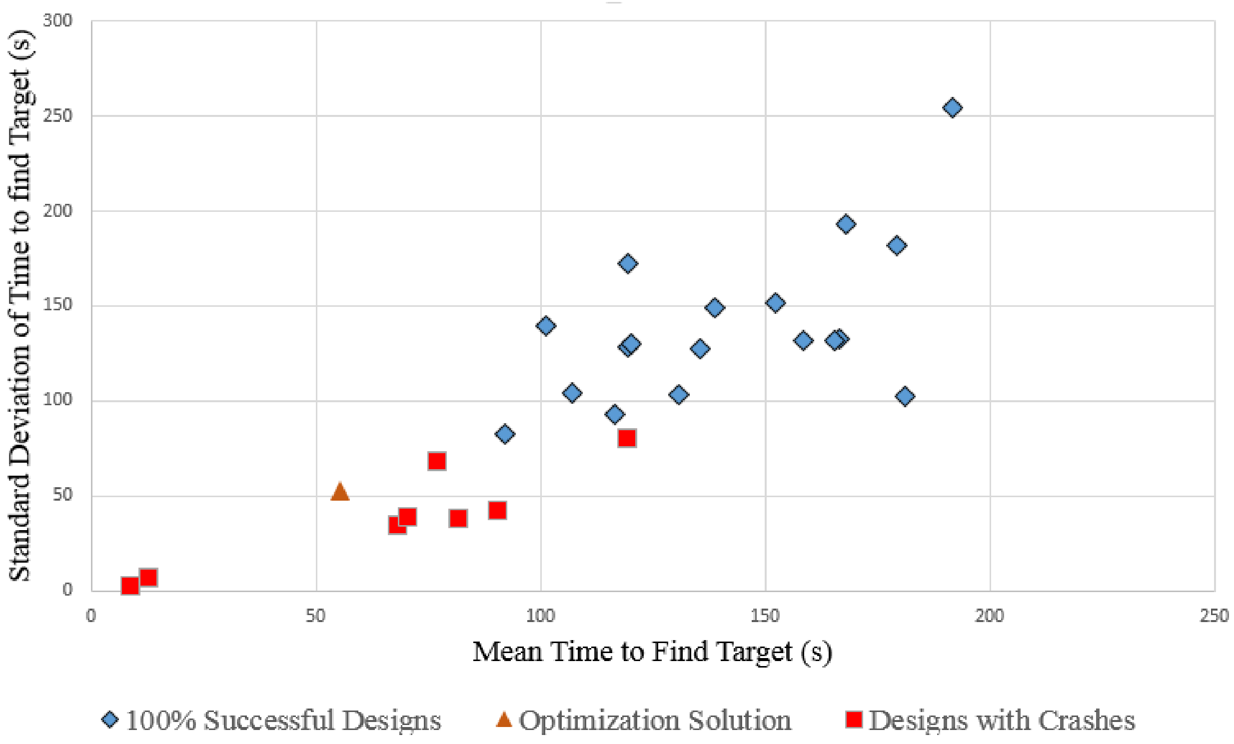


Fig. 5 Bi-Objective Plot of 100% Successful Designs (adapted from [7])

D.7. References

[1] Beyer, H.-G., and Sendhoff, B., "Robust optimization–a comprehensive survey," *Computer methods in applied mechanics and engineering*, Vol. 196, No. 33-34, 2007, pp. 3190–3218.

[2] Taguchi, G., and Phadke, M. S., "Quality engineering through design optimization," *Quality Control, Robust Design, and the Taguchi Method*, Springer, 1989, pp. 77–96.

[3] Paiva, R. M., Crawford, C., and Suleman, A., "Robust and reliability-based design optimization framework for wing design," *AIAA Journal*, Vol. 52, No. 4, 2014, pp. 711–724.

[4] Sahinidis, N. V., "Optimization under uncertainty: state-of-the-art and opportunities," *Computers & Chemical Engineering*, Vol. 28, No. 6-7, 2004, pp. 971–983.

[5] Ólafsson, S., and Kim, J., "Simulation optimization," *Simulation Conference, 2002. Proceedings of the Winter*, Vol. 1, IEEE, 2002, pp. 79–84.

[6] Carson, Y., and Maria, A., "Simulation optimization: methods and applications," *Proceedings of the 29th conference on Winter simulation*, IEEE Computer Society, 1997, pp. 118–126.

[7] Barton, R. R., and Meckesheimer, M., "Metamodel-based simulation optimization," *Handbooks in operations research and management science*, Vol. 13, 2006, pp. 535–574.

[8] Andradóttir, S., "An overview of simulation optimization via random search," *Handbooks in operations research and management science*, Vol. 13, 2006, pp. 617–631.

[9] Ólafsson, S., "Metaheuristics," *Handbooks in operations research and management science*, Vol. 13, 2006, pp. 633–654.

[10] Simpson, T., Mistree, F., Korte, J., and Mauery, T., "Comparison of response surface and kriging models for multidisciplinary design optimization," *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, 1998, p. 4755.

[11] Krige, D. G., "A statistical approach to some mine valuation and allied problems on the Witwatersrand: By DG Krige," Ph.D. thesis, University of the Witwatersrand, 1951.

[12] Koch, P. N., Wujek, B., Golovidov, O., and Simpson, T. W., "Facilitating probabilistic multidisciplinary design optimization using kriging approximation models," *AIAA paper*, Vol. 5415, 2002.

[13] Huang, D., Allen, T. T., Notz, W. I., and Zeng, N., "Global optimization of stochastic black-box systems via sequential kriging meta-models," *Journal of global optimization*, Vol. 34, No. 3, 2006, pp. 441–466.

[14] Tesch, M., Schneider, J., and Choset, H., "Expensive function optimization with stochastic binary outcomes," *International Conference on Machine Learning*, 2013, pp. 1283–1291.

[15] Boularias, A., Bagnell, J. A., and Stentz, A., "Efficient Optimization for Autonomous Robotic Manipulation of Natural Objects." *AAAI*, 2014, pp. 2520–2526.

[16] Crain, T., Bishop, R. H., Fowler, W., and Rock, K., "Interplanetary flyby mission optimization using a hybrid global-local search method," *Journal of Spacecraft and Rockets*, Vol. 37, No. 4, 2000, pp. 468–474.

[17] Yang, Z., Luo, Y.-Z., and Zhang, J., "Two-level optimization approach for Mars orbital long-duration, large non-coplanar rendezvous phasing maneuvers," *Advances in Space Research*, Vol. 52, No. 5, 2013, pp. 883–894.

[18] Bérend, N., Bertrand, S., and Jolly, C., "Optimization method for mission analysis of aeroassisted orbital transfer vehicles," *Aerospace science and technology*, Vol. 11, No. 5, 2007, pp. 432–441.

[19] Goulos, I., Hempert, F., Sethi, V., Pachidis, V., d'Ippolito, R., and d'Auria, M., "Rotorcraft Engine Cycle Optimization at Mission Level," *Journal of Engineering for Gas Turbines and Power*, Vol. 135, No. 9, 2013, p. 091202.

[20] Jones, B., and Nachtsheim, C. J., "A class of three-level designs for definitive screening in the presence of second-order effects," *Journal of Quality Technology*, Vol. 43, No. 1, 2011, pp. 1–15.

[21] Firth, D., "Bias reduction of maximum likelihood estimates," *Biometrika*, Vol. 80, No. 1, 1993, pp. 27–38.

[22] Myers, R. H., Montgomery, D. C., Vining, G. G., and Robinson, T. J., *Generalized linear models: with applications in engineering and the sciences*, Vol. 791, John Wiley & Sons, 2012.

[23] Carroll, R. J., *Transformation and weighting in regression*, Routledge, 2017.

E. LESSONS LEARNED WITH MODEL DEVELOPMENT KIT/DOCGEN

Author: Benjamin Kruse

Our sponsor liked the details on lessons learned that were discussed in using OpenMBEE, Model Development Kit (MDK)/DocGen, Model Management System (MMS) and View Editor. Many of these details and recommendations been provided during presentations and videos, and are summarize in this section of the report.

E.1. Introduction

OpenMBEE [1] stands for Open Model-Based Engineering Environment. As such it provides versioning, workflow management, and access control through its Model Management System

(MMS). The MMS stores the model data in a way that it is accessible by other tools that also utilize RESTful web services, e.g. the SysML [2] desktop client MagicDraw [3] through the Model Development Kit (MDK) plugin or the light-weight web-based View Editor, which requires models obeying the view and viewpoint paradigm. With this OpenMBEE aims to enable “multi-tool and multi-repository integration across engineering, computing, and management disciplines” [1], including the tracking of relations between such heterogeneous data sources.

The herein summarized lessons learned come from the currently used tools and versions, which are v. 18.5 SP3 for Teamwork Cloud (TWC) [4] as well as MagicDraw or Cameo Systems Modeler [5] and MMS v. 3.2.2 together with View Editor v. 3.2.1 and MDK v. 3.3.5. Since MagicDraw and Cameo Systems Modeler perform the same role of a SysML modeling tool, only MagicDraw will be mentioned further throughout this text. The use of OpenMBEE is also under investigation as part of the surrogate pilot study for the NAVAIR SE transformation framework experiment [6] and implemented on an Amazon Web Services (AWS) server.

An overview over the OpenMBEE implementation is given in Figure 81, showing the three main parts of OpenMBEE: MMS, MDK and View Editor in relation to MagicDraw as the SysML modeling tool, its models, a web browser, an xml editor and external access through the Interoperability and Integration Framework (IoIF) under development. More details about Figure 81 are explained within the following sections. Lessons learned about the MDK are presented in section E.2, focusing mainly on DocGen in section E.2.1. Section E.3 is about the MMS. It also includes its use through the MDK and MagicDraw. Section E.4 covers findings for the View Editor and the document ends with section E.5, mainly about TWC.

E.2. Model Development Kit (MDK)

The MDK is an API-based plugin for MagicDraw, to support building system assemblies through modeling augmentation and validation, enable syncing with MMS and using the DocGen language to create model-based documents using views and viewpoints. For this it has three main components: The Systems Reasoner, MMS Sync and DocGen. Further information can be found in its user guide [7].

As a MagicDraw plugin it contains its own profile, extending SysML together with its own code, as seen on Figure 81. To use the MDK plugin one must use the SysML Extensions profile with each SysML model, analog to other profiles, e.g. the one for SysML itself. Elements in the profile, whose name starts with “zz” are deprecated and should no longer be used. The plugin’s environment options are mostly for developers and do not need to be changed, unless e.g. a specific location for custom user scripts for DocGen is to be specified.

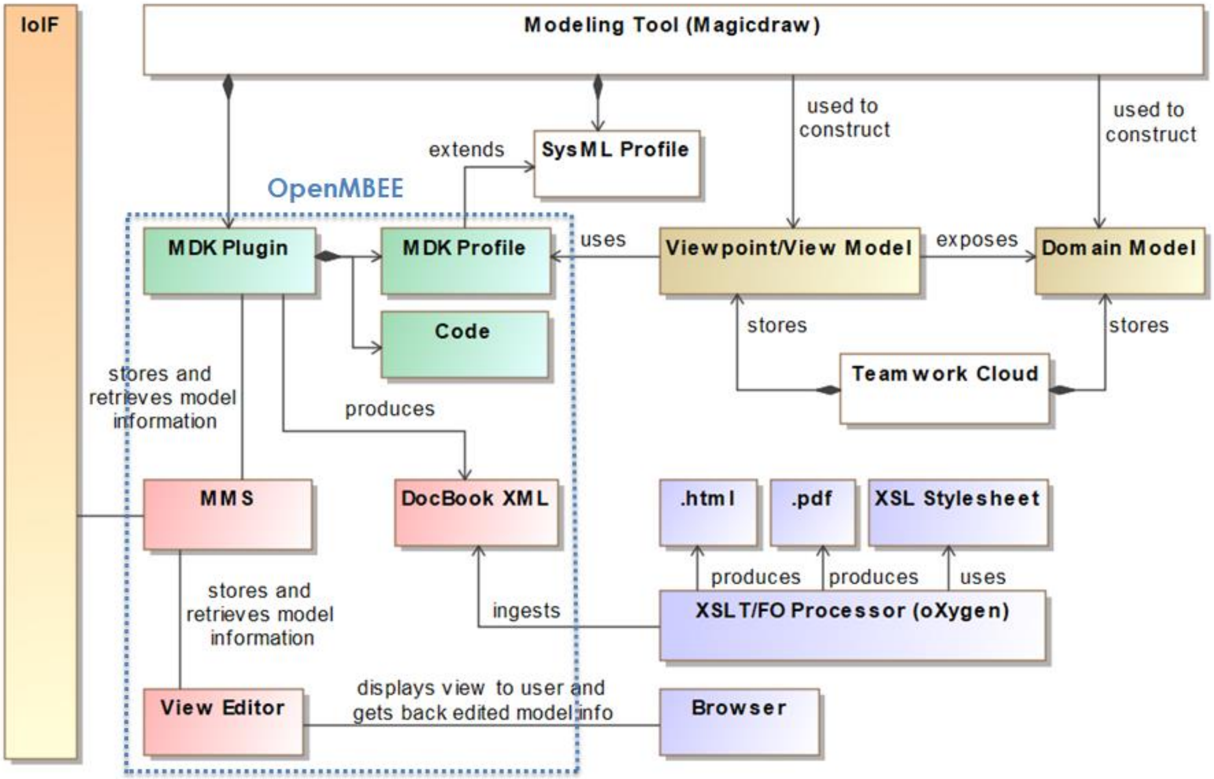


Figure 81: OpenMBEE overview (adapted from [8])

An important aspect of the setup of MagicDraw with MDK is the adaption of the magicdraw.properties file in the MagicDraw/bin directory: "-Dorg.osgi.framework.bundle .parent=ext" must be added into the JAVA_ARGS system environment variable definition. In case Magicdraw reverts back to offline mode after login because of active-mq, one must add the active-mq server host IP address running MMS into the operating systems hosts configuration file. For the AWS and Windows 10 this means to add: "ime.sercuarc.org mms-activemq" into the hosts file, for example on a Windows operating system, the file is typically located at: C:\Windows\System32\drivers\etc\hosts.

E.2.1. DocGen

DocGen is a realization of the view and viewpoint paradigm [8] for SysML. It allows one to parse and execute view and viewpoint models constructed by the provided stereotypes of the SysML Extensions profile by MDK, as seen in Figure 81. Further information is provided by the MDK DocGen User's Guide [9]. DocGen integrates into the existing SysML model and uses its semantics to indicate what of its content is to be exposed and how this happens.

E.2.1.1. View and Viewpoint Hierarchy

A generic example of a view and viewpoint model is displayed in Figure 82. To create such view hierarchies, the MDK's view diagram is used. A document element, which is a more specialized view, has associated views as its sections. The order of these sections is defined by the order in which the composition relations are created. To avoid changing a large number of these relations one should either change the order of the views in the View Editor or use a more hierarchical structure with views having other views as sub-sections.

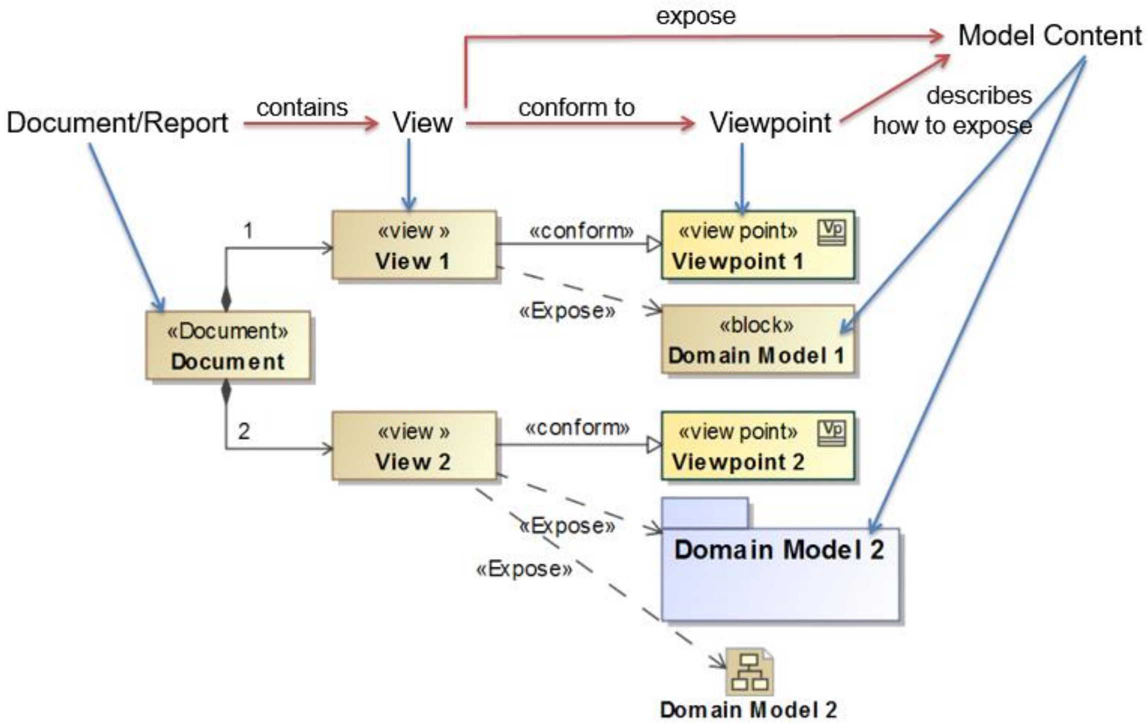


Figure 82: View and viewpoint hierarchy example

Document elements have additional properties to define their meta-information and the front matter, e.g. a footer, legal notice, involved institutes or companies and a logo. Please note that there are JPL/Caltech specific default values included in the stereotype, that potential logos are not included in generated PDFs using the Oxygen default DocBook PDF transformation [10], and that these front matter elements do not appear in the View Editor. The warning that a document element does not conform to any viewpoint that appears during document generation can be ignored.

Each view imports the model content that is to be exposed, using the Expose relation. Any types of model elements can be exposed, as seen on Figure 82, even the view element itself. The exposed elements are then passed along to the viewpoint, to which the view conforms. The conform relation is a more specialized generalization that causes the view to inherit the properties of the viewpoint. Among these properties, each viewpoint has a contained activity which is the behavior of the viewpoint. This is expected to change in MDK v. 3.4, where the viewpoint behavior must no longer be contained in the viewpoint to be consistent with the SysML specification [2] and its implementation in MagicDraw.

E.2.1.2. Use of DocGen

The views define the document elements and structure, and viewpoints essentially become document templates for exacting information from exposed views. "When one wants to generate a document from a model using a specific template, one can simply create a conforming view that imports the desired model elements as arguments to the template" [8]. If no such document template of views and viewpoints is used, one should first start with creating a document and its view hierarchy to reflect the planned sections and subsections of the document. In parallel the exposed model elements are added to determine the type of the exposed elements. Only then the viewpoints should be created or added from a library.

One can use DocGen to not only export and document model information, but also to guide the modeling and development process itself through required model elements fitting to pre-defined and reused viewpoints. They can for example check whether certain information exists in the model, whether it is complete or correct and enforce a uniform model structure to make the model content accessible for them in the first place. DocGen is also required for creating view instances for the View Editor, analog to directly generated documents. This further enables access to the model content for stakeholders not familiar with SysML and it can also serve as a pre-selection of model elements to read out of MMS. As done for instance in the demonstration of integrating SysML over OpenMBEE and MMS with the Semantic Web Technology (SWT) layer of the IoIF for visualization.

To create valid view and viewpoint hierarchies and generate DocGen documents in MagicDraw, the MDK's DocGen menu offers commands to validate the document, to check its viewpoint conformance and to generate a document preview for quickly checking views with their presentation elements and the target/text from the exposed model elements.

The generation of a DocGen 3 document creates a DocBook 5 xml file [11] together with an images folder that contains all exposed diagrams as png and svg files. These images get reused during each subsequent document creation. So they must be removed to get created again, e.g. after a change of the SysML model. The generated document is static and no longer linked to the MagicDraw model, unlike the same content in the View Editor. To create PDFs type files out of the xml file with its images Oxygen XML Editor v. 19.1 is used [10], as shown on Figure 81 in the lower right corner. Oxygen displays the plain xml text as well as an already formatted view in author mode and provides a default DocBook PDF transformation, which can be adapted or replaced by a custom stylesheet [9].

An initial library of reusable viewpoints is created for example for the surrogate pilot study in form of the IM90 model. Any model that contains view elements needs access to their exposed elements as well as the viewpoints, which both can be in separate used projects. Especially generic default viewpoints to expose recurring model elements that do not require custom stereotypes from the exposed model content are suited for such a library. Creating such a library in parallel to its use, it is important to estimate the impact of changes to such heavily reused viewpoints. Some included viewpoints are not for direct use linked to a view, but only to call their behavior from other viewpoints, e.g. to combine always the same table layout with different ways of collecting and filtering exposed elements. With the upcoming changes of MDK v. 3.4, where viewpoint behavior can be specified as any behavior and not only the owned one, the library will have to change to provide activities as generic viewpoint methods instead. This allows the actual used viewpoints to be application specific, e.g. with defined stakeholders or concerns according to the SysML v. 1.4 specification.

Findings from the surrogate pilot study show that not only viewpoints can be reused across projects, but the views can, too. This has the advantage that any additions or comments made to the view in the View Editor, which do not get into the MagicDraw model, still exist consistently everywhere the same view is used, no matter the actual document or model. Also, having the view hierarchy in a separate model that only uses the exposed domain model, allows users to have editing permissions for the document in the View Editor while only having reading permissions for the exposed underlying model. This allows commenting on the model without being able to change it directly.

From a separate small-scale test of DocGen for exposing elements from a DoDAF 2.0 model using the UPDM 2 plugin, the following lessons are learned. Having specific stereotypes can be beneficial for the creation of equally more specific viewpoints. They allow a more distinct checking for model elements for an improved modeling guidance and model checking, e.g. by checking if SysML non-native elements like a performer, its performing relation and involved scenarios do exist. An issue with the combined use of DoDAF/UPDM and DocGen lies in the both-sided, yet different use of view and viewpoint elements. These elements in DoDAF/UPDM are specialized packages that directly contain the related model elements, while the DocGen elements are specialized classes that allow composition and generalization relationships between them. That is why a combination of both elements cannot work, since stereotypes can only extend a single assigned metaclass [2]. This dual use of views and viewpoints results for example in DocGen views exposing DoDAF views as well as viewpoints, while needing multiple different DocGen viewpoints to expose all views within a single DoDAF viewpoint.

E.2.1.3. Viewpoint Creation

Creating a viewpoint, one has to create a viewpoint method diagram of MDK within it. This automatically creates the required activity, which describes the viewpoint behavior. The activity defines what to do with the exposed model content, by using MDK's stereotyped actions. This behavior has to fit to the type of the exposed model elements to create the intended results. A viewpoint that for instance collects elements from a diagram will not work when a package is exposed and a viewpoint that collects owned elements will probably not work when having a diagram as input. If requirements or blocks are supposed to be nested within each other or only within packages, collect them all with the used viewpoints.

A viewpoint should follow the general process of first collecting elements, then potentially filtering and sorting them, before concluding with the description of how to present them. A potential recursive call of the activity to iteratively repeat the process may follow at the end, too. Testing viewpoints with views that expose a limited set of elements as well as using the DocGen preview function is especially useful for larger models. The following sections describe lessons learned with respect to the use of the provided stereotyped actions of the viewpoint method diagram for creating viewpoint methods. Examples for such diagrams are displayed in Figure 83 and Figure 84.

E.2.1.4. Collecting Elements

Starting with the actions to collect elements there is often a depth of the collection to specify, as seen on Figure 83. It determines the level up to which elements are collected. For example, when collecting owned elements with depth=1, only the directly owned elements are taken. Depth=2 would mean to not only collect the directly owned elements but also all elements they own directly. Special notice for depth=0, which stands for infinite collection steps. This can be computationally expensive for larger models.

The action to collect elements from diagrams also works for tables, but not for matrices. Also, for tables the collection does not include elements that got removed from the table, but it does include all elements that were filtered from it in MagicDraw. This makes this MagicDraw search mechanism for selecting very specific elements in MagicDraw not usable for DocGen, analog to the elements in smart packages, which cannot be collected, either.

Collecting the types of elements works for e.g. part properties, which have the connected block as type. The type of the block itself does not return the block stereotype or the class metaclass from UML. Collecting by associations differentiates between aggregations, compositions and unspecific associations. This works as intended for blocks, but not the generic associations between use cases and actors. Those must be accessed using Object Constraint Language [14] OCL expressions, as described in section E.2.1.8.

E.2.1.5. Filtering and Sorting Elements

All filter and sort actions can reverse their results, i.e. to exclude elements instead of including them or to switch the sorting order. It is not required that elements fulfill all criteria when filtering for multiple at once, e.g. when filtering for multiple diagram types with one action. To filter for multiple criteria, multiple filter actions must be used. Also, when filtering for specific diagram types there has to be differentiated between the UML diagrams and their SysML counterparts. The two actions filtering for specific metaclasses and stereotypes, which are used in Figure 83, are also useful to filter out elements in the MagicDraw model that are not displayed in its containment tree and which should not get included in the document. Filtering for requirements only works by filtering for their stereotype and not for their metaclass, which can be defined in a filter action, too.

Finally, sorting elements according to their names happens through the SortByAttribute action with the name as the desired attribute. The sorting order can be unintuitive, since e.g. Req-100 to Req-126 will come before Req-16 and its following ones. Elements with the name or id Req-016 would be sorted correctly. Empty spaces are taken into consideration at sorting, too.

E.2.1.6. Dynamic Sectioning, Structured Queries and Dynamic Views

To structure the viewpoint method diagram as well as the generated view, there exist structured query actions, which contain further nested actions. Their first main purpose is the dynamic creation of subsections within a view. The right structured query element on the left side of Figure 83 for instance loops through all originally exposed elements and creates a subsection for each of them, while automatically assigning section titles that follow the names of the exposed elements. This is especially useful when e.g. creating subsections for each diagram, which allows them to be named accordingly.

The second purpose of structured queries is to allow the re-access of the originally exposed data without it being impacted by the included actions, e.g. to no longer be limited by the filter for class typed elements in the structured query on the left. The input elements are always identical to the output elements for structured queries as well as tables.

An alternative way of handling the flow of exposed elements is displayed in Figure 83 on the right side: A parallel fork node is used together with a union merge node. This allows to have all originally exposed elements together with additional derived elements, which are all their owned elements in the example on Figure 83. The decision node cannot be used for viewpoint methods. Its behavior can e.g. be replaced by adequate filter actions.

To further refine the behavior of viewpoint methods as well as to support their reuse, call behavior actions can be defined. They need the dynamic view stereotype, which provides similar properties as structured queries, e.g. to loop through all elements individually or to automatically assign section titles. Furthermore, they have the setting to skip the execution, if the action input is empty. Yet, this does not work properly for the view instance generation for the View Editor. Viewpoints method activities can also call themselves, e.g. to recursively loop through a structure of nested packages while exposing their content.

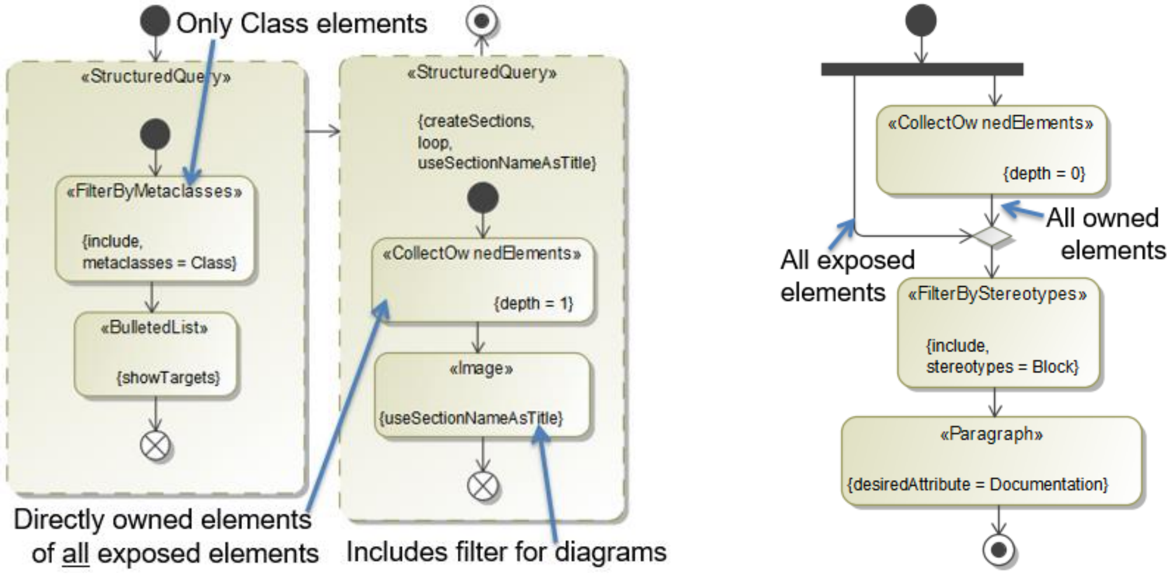


Figure 83: Viewpoint methods using structured queries (left) and parallel and merge nodes (right)

E.2.1.7. Presenting Model Data

Several ways exist to present the collected, sorted and filtered model elements. First there is the possibility to create images of all exposed diagrams, matrices and tables using the Image action. These actions already include an internal filter to only consider valid elements. Their generated figures are not editable. This will improve with MDK 3.4 and MMS 3.3 that will include an integration of Tom Sawyer Perspectives [12] for dynamic visualizations in the View Editor. Especially when not only using the View Editor but also printable versions, e.g. PDFs, it is important to limit the size of the diagrams within the model, to have them actually fit into the generated report. Also, when exposing tables their overall size does not change when done so in MagicDraw, e.g. after making row or columns headers bigger. The title of each figure can again be automatically assigned as the name of the section, which can be the name of the diagram, if it is used e.g. to automatically create sections using a structured query. Each image also gets its documentation added below it in form of a paragraph. Simulation plots of the Cameo Simulation Toolkit [13] cannot be directly exposed as images.

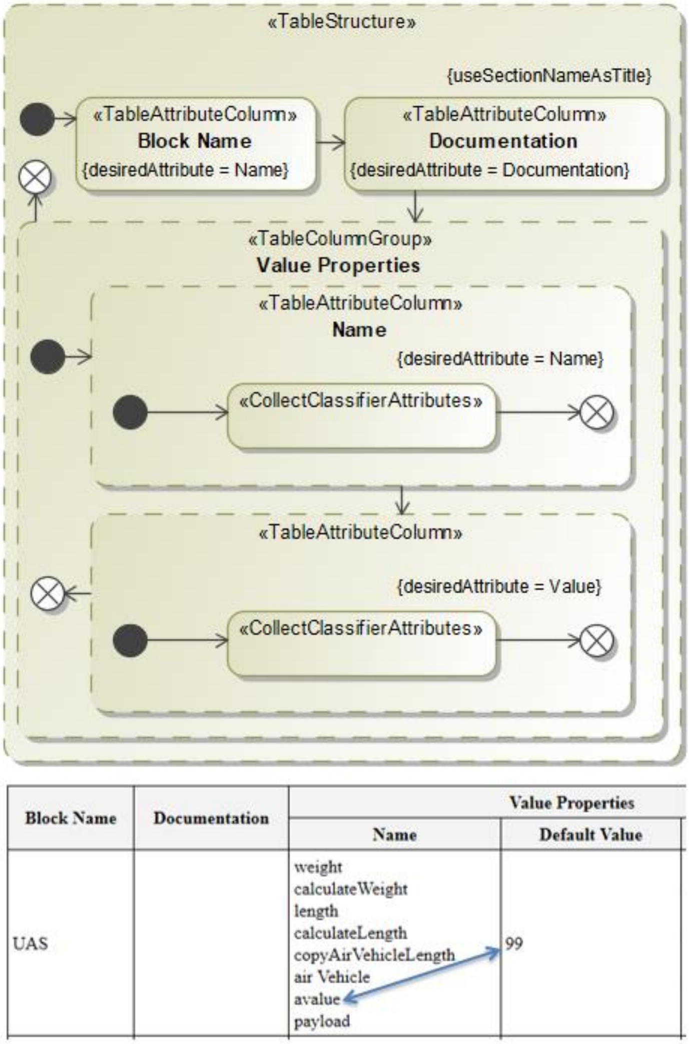
The second presentation element is the paragraph. It creates paragraphs of text for either the name, documentation of value of the exposed elements. One of these three attributes must be selected, even if there is an alternative fixed output text specified in the action's body property. This can also be used for OCL expressions to be evaluated, e.g. to display different text depending on the exposed model content. Potential html-formatted text is partially kept. Included figures for instance are not found to be inserted. When generating xml files, the documentation of each view is automatically included in form of a paragraph. This is not the case for the View Editor and since the feature is considered to be deprecated it should not be used. Instead such text should be included in exposed model elements, e.g. the package that contains the relevant model content, the documentation of diagrams or particular comment elements. To define the formatting of the exposed text in the xml files, yet not in the View Editor, certain expressions can be used. For example, <emphasis role="bold"> and </emphasis> can be added to mark the text in between as bold. Similar expressions can be identified by changing a text field in the View Editor and updating this change to MagicDraw, resulting in such additions instead of, e.g. an html formatting.

Third, bulleted list can be created. They can either use simple bullet points or be numbered. They can expose the name of the elements when showTargets is set to be true, as shown in Figure 83 on the very left. The documentation of each element can be shown, too. The feature to show stereotype property names does not work.

Finally, there exist multiple ways to create tables using DocGen. The two easiest ways to create tables are the actions GenericTable and PropertiesTableByAttributes. The generic one recreates every input table or matrix. It does not work with other exposed elements, but it is the only way besides the Image to present matrices. The relations in matrices are hereby represented as hooks in the View Editor and as question marks in xml. When using it to expose multiple tables at once, it has a bug that causes it to add up the number of columns for each following table. This can be avoided by using it within a structured query that loops through each table individually. The PropertiesTableByAttributes action is specifically for the properties of stereotyped elements, e.g. to create a table showing id and text, as defined by the requirement stereotype.

The manual way to create custom tables uses the TableStructure action to contain nested actions defining each column. An example is given in Figure 84. Rows are created for each exposed model element that the table action gets as input. Rows and columns can be switched by using the Transpose property. Tables have the property to loop through each exposed element analog to structured queries. This creates a table each and again allows to automatically name them. To create columns, there exist three different types of column actions for exposing an element attribute, i.e. name, documentation or value, for exposing an element property, e.g. the requirement id analog to the PropertiesTableByAttributes, and for exposing according to an OCL expression. Each column header is defined by the name of the used column action, as it is displayed in Figure 84 with the generated result below. To create further nested collecting and filtering, each of these three actions exists also as a structured activity node to contain nested actions, analog to the structured query. Such attribute columns that collect the attributes of the exposed blocks are used for instance in Figure 84 as part of a column group, which further structures the table.

When using the View Editor, it is recommended to limit the number of shown properties per row, to avoid shifted elements when exporting it. This is displayed on the bottom of Figure 84 with the “99” being the default value of the “avalue” property. Alternatively, only a single value property should be in each row, e.g. by creating smaller individual tables for each block, i.e. for the shown UAS. Further best practices when creating tables include the use of blocks with their value properties to specifically capture entries for each column of the tables. This way it is possible to create custom table entries, including figures by using the Image action within a structured query that defined the column. Finally, it can be necessary to split up a table with many broader columns into multiple separate tables to keep each of them narrow enough to fit on a printed page, if that is required.



Block Name	Documentation	Value Properties	
		Name	Default Value
UAS		weight calculateWeight length calculateLength copyAirVehicleLength air Vehicle avalue payload	99

Figure 84: Partial viewpoint method for creating a table (top) with its result exported from the View Editor (bottom)

E.2.1.8. OCL and User Scripts

As mentioned earlier there exist actions to collect, filter, constrain and present model elements based upon OCL queries, too. OCL, the Object Constraint Language [14] of the OMG is a declarative language for rules that can be applied to SysML models. Entered into the viewpoint method actions these expressions can either be applied to the whole set of exposed elements or iteratively to each single element.

Part of the MDK plugin is the OCL Evaluation tool. It is very useful to test OCL expressions on elements from the containment tree or an open diagram before using them in a viewpoint. Entering the query “self” for example shows the selected elements themselves including a list of potential completion choices that could be added to the query. The expression “self” is an OCL keyword for referring to the current instance of the context of the query [14], i.e. the selected elements. The tool is also useful to identify potential hidden elements in the models that are not shown in the containment tree, but get exposed by a view. Occasionally, its result can vary from the execution of the same OCL expression in an action. This can happen when a result from an OCL operation needs to be cast (also called: re-typed) as the correct element type that is expected from the query in order for further operations to be carried out [9]. The expression “self.r()” for instance works as query to access the relations of exposed elements, but used in an action it requires the expression “self.oclAsType(Class).r()” to work. This casting also allows object to be re-typed as another type and to use a property of an object defined on a subtype of the currently known type of the object.

The following examples should be seen in addition to the DocGen User’s Guide [9]. When filtering elements per name, any name is valid for the query “self.+”, while e.g. “UAV.*” returns all elements whose name starts with “UAV”. The expressions “self->notEmpty()” and “self->size() > 0” both check whether the set of input elements in empty, which can be used to show a warning message if no valid model elements are found. To filter for element types, expressions similar to “oclIsTypeOf(Class)” and “oclIsKindOf(NamedElement)” can be used to determine if the elements are instances of the given type, i.e. if they are exactly of the type “Class” and if the elements conform to the stated type, i.e. if they are any named elements. An example that uses such an expression within an if clause is: “if self.ownedElement.oclIsTypeOf(Class)->includes(true) then true else false endif”. It tests if any class type element is owned by the selected elements. To expose associations between actors and use cases the expression “self.associationOfEndType” can be used. “self.memberEnd” then allows access on the connected elements at each end of the association. The connected elements of any type of dependencies can be access by “self.client” and “self.supplier”. To extract the multiplicity of elements, e.g. associations, ports or pins, the two OCL expressions “self.lower” and “self.upper” are used.

When exposing instances with their values one has to access their slot elements, which are derived from the value properties of the blocks that are the type of the instances. Therefore, these slots do not have a name or type of their own, but only the name and type of their value property as the defining feature, which requires the expressions “self.definingFeature.name” and “self.definingFeature.type”. The resulting representation of these exposed elements is not editable in the View Editor. This is because of the more indirect access through the OCL expressions in contrast to e.g. the values of the slots, which can be accessed directly through an attribute column and then edited in a table in the View Editor.

To allow even more powerful operations during the document generation it is possible to define user scripts, e.g. in Jython, which are linked to actions on viewpoint method diagrams. For instructions as well as an example about collecting elements see the DocGen User’s Guide [9]. In the surrogate pilot study such scripts are used to automatically create textual statements about required system states and transitions, derived from state machine diagrams.

E.2.1.9. Additional Features

Viewpoint methods have the capability to start Cameo Simulation Toolkit simulations when being executed. For this they have to include the Simulate action and their view has to expose a SimulationConfig element. The simulation requires auto start, automatic end and it must not need any manual input. The DocGen view generation terminates after 60 seconds with a timed-out simulation run. Starting simulations through DocGen is intended to be used to e.g. recalculate values after changes in the View Editor. This works only when it is possible to run jobs in the View Editor, which is currently not the case, since that required software is not released. When using the Simulate action from within MagicDraw new simulation results, e.g. in the form of created instances, are not collected during the same document generation as the simulation.

Other features that do not yet work as intended or could not be tested successfully are the Plot, TemporalDiff and ViewpointConstraint actions. Plot is supposed to graphically plot exposed data, e.g. in form of a radar or parallel axis plot. TemporalDiff is supposed to display the difference of an attribute of an element at two points in time from two ISO 8601 timestamps. It is only supposed to work with the View Editor, where such element history comparisons of single elements are already possible. The ViewpointConstraint action is supposed to allow a constraint to be evaluated at any point in a viewpoint method diagram on any elements passed to the action.

Related to DocGen as well as the Systems Reasoner is the Constraint stereotype in the MDK plugin. It also does not work as intended for both of its two applications. When attaching the stereotype to a viewpoint method action or a comment linked to one it does not give any result during document generation. When validating it within a comment attached to any element, one can get a growing list of errors with each manual validation even for true constraints that should give no error at all.

E.2.2. Systems Reasoner

The MDK Systems Reasoner’s purpose is it to support building system assemblies in SysML. It also comes with its own documentation, the MDK Systems Reasoner User's Guide [15]. The model validation feature of the Systems Reasoner for instance checks whether all inherited properties are set to be redefined in the model or if instances miss slots for the value properties of their blocks. As an alternative to MagicDraw’s Excel Import plugin, the Systems Reasoner also has the capability to use CSV tables to create multi-level hierarchies of blocks with their value properties that include specialization relations to generic parent elements. Another feature is the quick one click creation of instances, as an alternative to the multiple steps of the MagicDraw Automatic Instantiation window.

A main function of the Systems Reasoner is the augmented creation of specialization trees. Multiple different commands exist for this, to either Specialize Structure, Specialize Structure Recursively or Specialize Recursively & Individually. The first command creates a single specialized block with all inherited elements ready to be redefined. Its part properties refer to the parent element’s part property blocks. The second command to Specialize Structure

Recursively is displayed in Figure 85. The left side shows an example system structure before the execution and the right side show it afterwards. The recursive specialization means that the blocks of part properties also get new child elements for further refinement. This supports for instance the progression from a logical architecture to a more detailed physical one, while maintaining full traceability between the layers of abstraction through the generalizations. Finally, the third command additionally creates individual child elements in the case that there are multiple part properties pointing to the same block in different roles.

Another similar function is for adding and realizing aspect relations. It creates a special aspect dependency as well as its realization through an associated child element of the aspect block. The purpose for this pattern is it to offer an alternative to using custom stereotypes with tagged values to define that certain types of elements have specific properties. These properties are instead inherited and can then be redefined. Occurring issues with using tagged values are for instance their representation on internal block diagrams or the definition of slot values of instances.

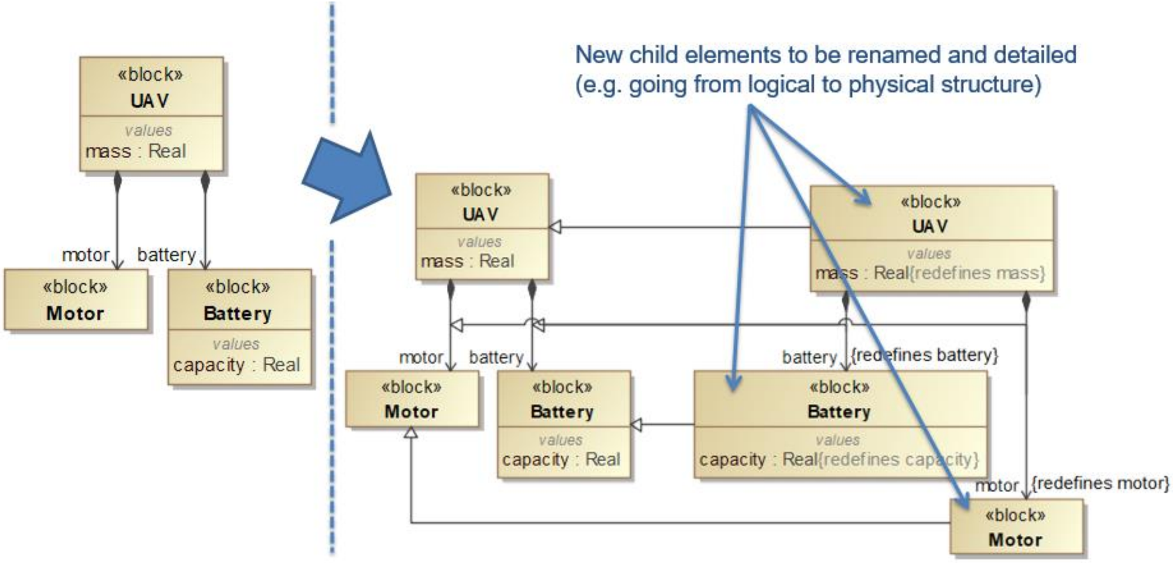


Figure 85: Example for specializing structure recursively, before (left) and after the execution (right)

E.2.3. MMS Sync

MMS Sync is the part of the MDK plugin that allows to commit models to MMS and receive updates from it. For further information about MMS, see section E.3. Since MMS Sync can only be used in conjunction with MMS, it is explained in section E.3.1 together with the use of MMS.

E.3. MMS

The MMS is a version control system for structured data, including versioning, workflow management, and controlled access through RESTful web services. It is intended to be a central data hub to facilitate multi-tool and multi-repository integration across engineering, computing, and management disciplines, by storing its data tool-independent and accessible. Here it is used to only store SysML model data from Magicdraw, e.g. for the View Editor. For this it captures all model elements of SysML projects, e.g. classes, instances, properties, values, relations, etc. including their change history and the generated view instances for the View Editor. Not saved is information about elements on diagrams with their layout. Therefore, it cannot be used to perfectly synchronize individual local copies of SysML projects. For the collaborative work on shared projects in MagicDraw, TWC is still needed.

The used MMS is implemented within Docker [16] to allow easy, consistent and platform-independent deployment. Docker performs virtualization of the operating system through so called containers. There are four containers for MMS: For Apache Tomcat [17], the Elasticsearch [18] search engine, the PostgreSQL [19] object-relational database and the Apache ActiveMQ JMS (Java Message Service) broker [20]. Within Apache Tomcat runs Alfresco [21] with the Alfresco Share web portal, the Alfresco repository as well as the View Editor.

The data is stored in MMS as JSON (JavaScript Object Notation) [22] objects that contains all individual model elements from MagicDraw with their IDs, names, types, values, etc. as well as additional information, e.g. project IDs, branch IDs or commit IDs. This allows MMS to capture the full element history with branches and tags, additionally to TWC. The exposed diagrams for the View Editor are saved as png and svg images in the Alfresco repository. Further files, e.g. attached files from MagicDraw projects can be stored there manually through Alfresco Share, too. Attached files in MagicDraw are a feature currently not supported by OpenMBEE, causing MDK to display validation errors for each attached file.

The access to the data is controlled through the user management with its roles and groups in Alfresco. It is possible through MMS's REST API, e.g. by the MDK or the View Editor, together with ActiveMQ for its live updates. Other tool-specific plugins similar to MagicDraw's MDK exist for "Mathematica, MATLAB, ModelCenter, and a AsciiMathML based expression editor" [1]. A first proxy for the IoIF, exists, too. That one uses the improved API of the upcoming MMS v. 3.3, which can be access through SwaggerUI.

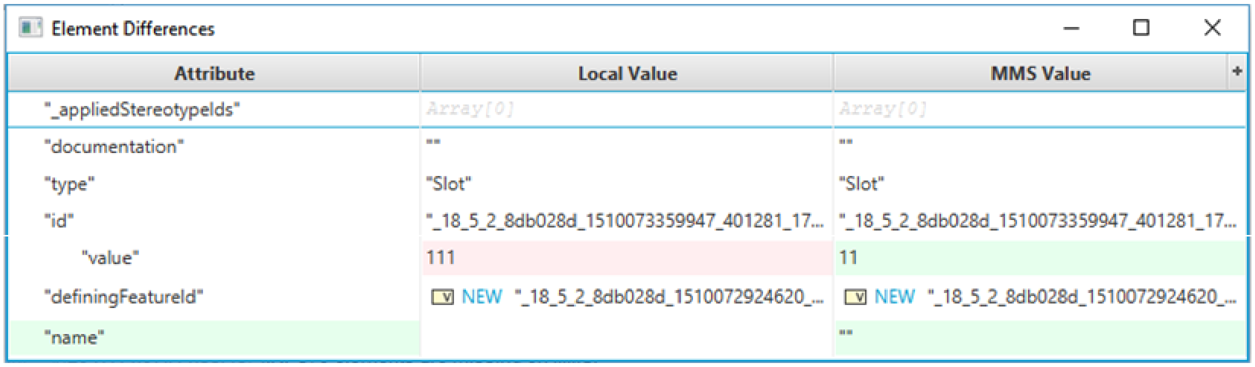
E.3.1. Use of MMS with MagicDraw

To use MMS from MagicDraw through MMS Sync, one has to apply the ModelManagementSystem stereotype from the MDK's SysML Extensions profile to a projects main package element. Then one can define its MMS URL property, e.g. http://ime.sercuarc.org:8081/ for the AWS MMS, and login. Known errors during MMS login and validation are described together with their solutions in the system setup of the MDK plugin in section E.2.

To commit a project for the first time one has to do a first MMS validation and then select to commit project and model at the validation result that states it as missing on MMS. This is followed by a window to create a new MMS org or select an existing one. The org stands for organization, to group projects together and simplify the user permission management in Alfresco, where an analog element, there called site, is created. To create a new org/site the user needs Alfresco manager permissions, to avoid a broken model in MMS. This potential issue is supposed to be fixed with MMS v. 3.3. Also, with the current MMS 3.2.2 deployment in Docker it requires that the admin password is set to its default, since it is included in the MMS build itself. After this first commit, one should follow with a second model validation and then update the model from MMS, to create a Holding Bin package and avoid future validation errors caused by it. Finally, a third validation together with a commit to MMS finishes the process. To initialize a single project multiple times one can reset the project ID as well as the element IDs in MagicDraw.

In general, when working with MMS and MagicDraw there are two different ways to synchronize the model data: The manual sync and the coordinated sync, which is specifically for TWC projects. The manual sync should not be used for TWC projects, despite to re-establishing parity in case the automatically executed coordinated sync fails. This is for instance the case for the project initialization with its holding bin, as described above, or it is required to update information in MMS about used projects changes.

The manual sync can be done for the whole model, the model up to a specified depth and for single elements only. Its results display the elements missing in the MagicDraw client, the elements missing on MMS and the elements, which are not equivalent between the two. If one has to validate manually, it is best to follow the CRUD rules about order of operations: 1) create 2) update 3) delete [7]. To resolve changes for those elements that are not equivalent between MagicDraw and MMS, one can display the differences for each element based on the JSON data schema. This is shown in Figure 86. Then the user can either commit it as it is in MagicDraw to MMS or update the element in MagicDraw as it is in MMS. If neither of the two possibilities is done, the element inconsistency will reappear at following validations.



Attribute	Local Value	MMS Value
"_appliedStereotypeIds"	Array[0]	Array[0]
"documentation"	""	""
"type"	"Slot"	"Slot"
"id"	"_18_5_2_8db028d_1510073359947_401281_17...	"_18_5_2_8db028d_1510073359947_401281_17...
"value"	111	11
"definingFeatureId"	NEW "_18_5_2_8db028d_1510072924620_...	NEW "_18_5_2_8db028d_1510072924620_...
"name"		""

Figure 86: Element differences of slot element with changed value and created name property

The coordinated sync does only work for TWC projects. It is triggered automatically at each TWC commit to ensure maintained consistency between the TWC model and the MMS model and to make further manual activities in form of the manual sync unnecessary. It syncs all changes that are tracked towards the model on TWC. Changed elements in MMS are updated into the MagicDraw model as possible. The resolving of conflicts, e.g. caused by changes to elements in both MagicDraw and MMS occurs analog to the manual validation through the then appearing window with validation results. A consequence of the coordinated sync is that the commit to TWC also requires a login to MMS. An overview over the tracked element changes can be seen any time in the Sync Status window in MagicDraw. It displays the local as well as the MMS changes.

TWC projects with branches can be committed and synced to MMS. This can happen analog to normal models or through an additional command in MagicDraw to validate the branches, which allows e.g. to commit branches to MMS without having to open them individually. It is important to note that while branches can also be created directly in MMS through the View Editor, a synchronization between MagicDraw and MMS is only possible for branches that were first created through MagicDraw.

Both ways to synchronize the model data do not include the (re-)generation of views for the View Editor. This must be performed manually, either for all views at once through the button on the top menu bar or for specific views only, with or without recursively including their subviews. Exposed images are hereby not only created within Alfresco, they are also created as svg files in a folder called “mdkimages” in the current user’s files. View instances must be generated for all views and documents before they can be seen in the View Editor. All changes to the model that e.g. introduce new exposed elements or impact exposed diagrams, require a repeated view generation. Changes to already exposed model elements, e.g. a changed value or documentation does not require a repeated view generation. Viewpoints that use OCL expressions almost always require the regeneration of views for any changes to be visible in View Editor. Views from used projects in the view hierarchy view instances must be generated in the main model. Despite the warning that the used view is in a module and was therefore not processed. In general, this should be done after updating the used projects in MagicDraw.

Besides the validation errors that can be caused by a missing setup of the holding bin or that are caused by attached files in MagicDraw there are some additional occurring validation results to be mentioned. Non-resolvable validation errors occur when changing a slot element in the View Editor, since that introduces an empty name property in MMS. This is shown in Figure 86. Yet, slots are not named elements in SysML, where consequently such a property cannot exist. These errors are supposed to be fixed in MMS v.3.3 and should be ignored for now, analog to validation errors regarding mounted, i.e. used projects. To resolve those changes in the used projects would be required, which includes changes to standard profiles. This type of error is solved with MMS v. 3.2.4. More harmful is a current java error preventing the use of other projects. It supposedly is caused by a feature of MMS to maintain traceability at element refactoring by keeping element IDs, which is in conflict to MagicDraw, where refactoring an element and changing its type also changes the ID of the element. Initial attempts to resolve these errors could not find such ID errors. The issue is still under investigation.

One final best practice is about the handling of used profiles and libraries in MMS, e.g. for stating units or element types, which are displayed as not found on the right of Figure 88. Generally, profiles and libraries should not be seen as individual models in the View Editor, but their model elements should still be available to get exposed. The recommended solution is to create a separate org for them that has only the system administrator as its member. Every user is then given read-only access of the individual projects in Alfresco Share by using the predefined user group “everyone” with the role Consumer. More information about this follow in section E.3.2. To commit the standard profiles of MagicDraw to MMS without changing them, it works to use a copy of them that is set up to use the SysML Extensions profile and given the MMS URL instead.

E.3.2. Alfresco for MMS

Alfresco [21], in its Alfresco Community Edition is an open-source content management system. It includes a central content repository and a web interface named Alfresco Share, which is displayed in Figure 87. It can be opened through the DOCLIB button in the View Editor, e.g. opening the URL http://ime.sercuarc.org:8081/share for the AWS OpenMBEE. Alfresco is used in

MMS to manage user accounts, their roles and permissions for models and branches, as well as the images for the View Editor. The possibility to use MMS with an alternative, similar tool instead of Alfresco is under consideration.

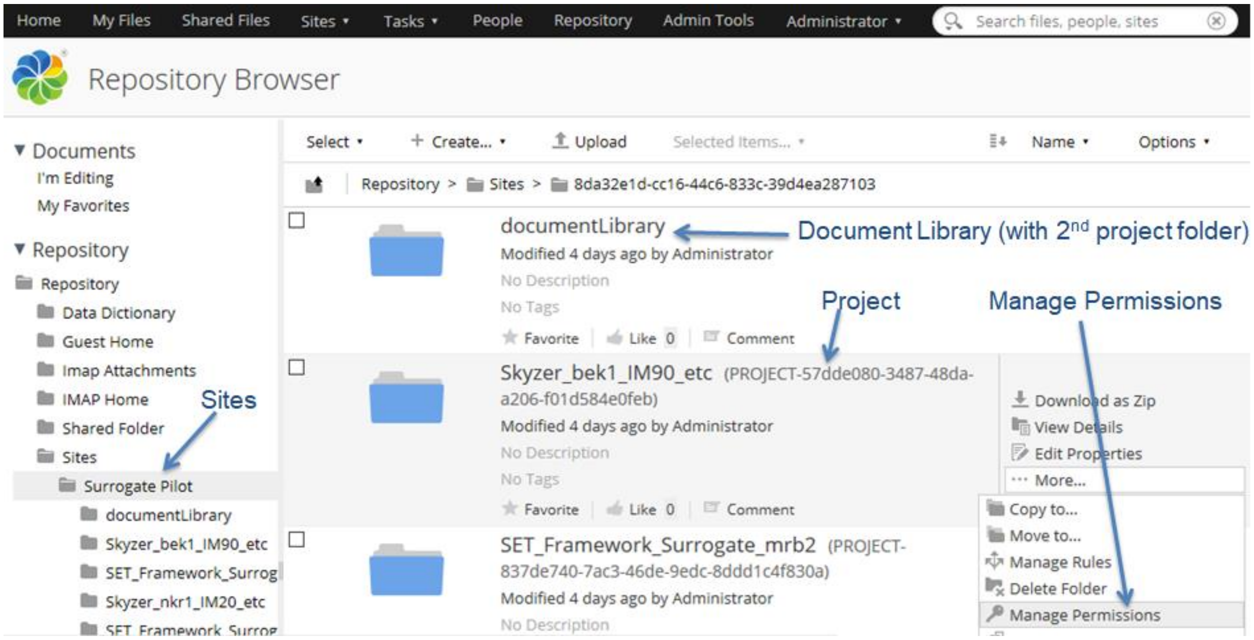


Figure 87: Alfresco Share displaying the repository with projects of the "Surrogate Pilot" site

The sites created in Alfresco through MagicDraw are project areas to share content among site members. They can be private, public or moderated. Moderated means that their existence is visible, but they require an invitation to participate. Each site has its customizable site dashboard and other features. There can be for example a calendar, a wiki, discussion boards, blogs and data lists, e.g. for issues or tasks.

In the Alfresco repository section each site includes folders for its projects, their branches and tags as well as other related documents. These folders of the Alfresco repository are also used to define the project, branch and tag specific user permissions in addition to the by default inherited overall site permissions for individual users or user groups. The important part is hereby the permission to the project folder within the site folder, as shown in Figure 87. This folder also contains the branches and tags. The analog second project folders in the document library are empty and can be used for other project-related files. When defining user roles for branches or tags one must first determine their ID, e.g. from the View Editor, since they are displayed as such and not by their name in Alfresco. Also, to access a branch or tag in the View Editor the user needs at least the Consumer role for the main model to access the branch or tag.

User accounts and groups are created in the Admin Tools section of Alfresco Share. Default user roles in Alfresco are: Managers, Collaborators, Contributors and Consumers. By default Managers have full rights to all site content, Collaborators have rights to edit but not delete content created by other site members, Contributors cannot edit or delete content created by other site members and Consumers have read-only rights in a site, and they cannot create their own content [23]. Yet, when using these roles for OpenMBEE, Contributor cannot even edit their own models in the

View Editor and Collaborators can delete content from other users in the View Editor, e.g. views. Therefore, one should probably avoid the use of Contributor.

Final findings regarding the use of Alfresco for OpenMBEE are first, that it is highly recommended to initially plan the sites and user groups to simplify the role assignment as much as possible, e.g. by using inherited permissions to avoid assigning individual project-level permissions for each new user. Second, there exists a known error in Alfresco that can revert changes to project permissions back by re-adding previously removed inherited roles, resulting in unwanted access and hence a security issue. Third, user email notifications are currently not implemented.

E.4. View Editor

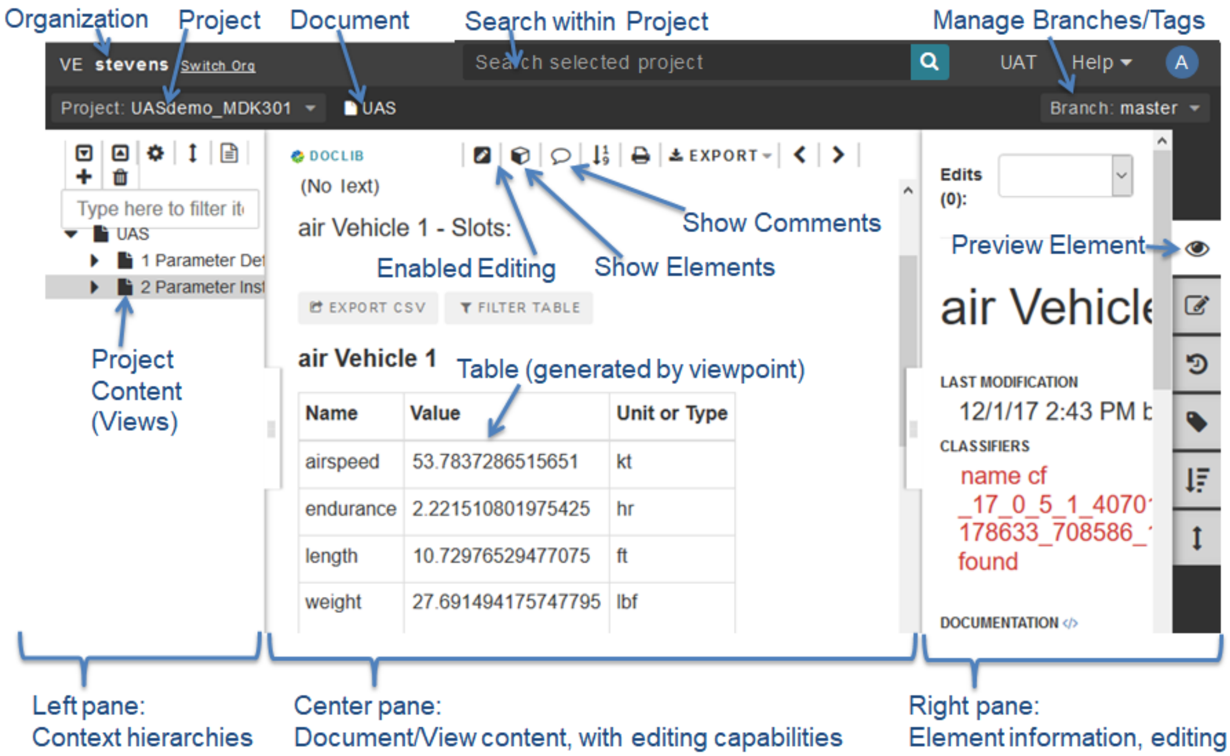
The View Editor is a light-weight web-based client to directly access and edit model data in MMS, based upon view instances that follow the view and viewpoint paradigm. Here it provides access to and interaction with consistent data from MagicDraw SysML models without having to use MagicDraw itself. This way the View Editor aims to improve communication especially with stakeholders that are not familiar with SysML via live documents and data of an authoritative source of truth, the MMS. It displays views that expose SysML model elements by conforming to viewpoints. As a part of OpenMBEE the View Editor also comes with its own user guide [24].

Potential use cases for the View Editor include the model-based generation of documents for reports as well as to document the model creation itself. For reports it allows not only the direct editing of the model, but also the addition of comments, the inclusion of links to external data and other documents as well as the addition of presentation elements beyond the ones generated from SysML, e.g. for videos. Tags can be created to permanently save the state of the document in the View Editor at a specific point in time and the history of single model elements can be traced and compared. This is also useful to capture the development of the underlying model by using generic and recursive viewpoints to expose all major modeling elements and diagrams, providing a quick overview over the state of the model.

E.4.1. View Editor Overview

With the MMS implementation in Docker that has the View Editor together with Alfresco, the URL for the AWS View Editor for example is: http://ime.sercuarc.org:8081/alfresco/mmsapp/mms.html. The login uses the Alfresco user accounts, as described in section E.3.2. The same user account also works for the public View Editor of OpenMBEE [1]. After the login the user has to select an organization and a contained project.

An overview of the View Editor window is given in Figure 88. On top there are general buttons for instance for the opened organization, project and document. The uppermost search window allows searching for all model elements in MMS that are in the opened project, its branches, tags and all used projects. Search results are displayed together with their element properties, including e.g. where they are from. Improved filtering for the search is coming with View Editor v. 3.3 for MMS v. 3.3, e.g. to collect only elements with a certain stereotype applied.



(No Text)

air Vehicle 1 - Slots:

air Vehicle 1

Name	Value	Unit or Type
airspeed	53.7837286515651	kt
endurance	2.221510801975425	hr
length	10.72976529477075	ft
weight	27.691494175747795	lbf

air Vehicle

12/1/17 2:43 PM b

name of _17_0_5_1_4070 178633_708586_ found

Figure 88: View Editor overview with top, left, center and right pane

The left pane shows the content of the opened document or view in form of a tree, which can be changed and filtered. The center pane is the main window, showing the opened view with its exposed presentation elements, e.g. a table for an air vehicle instance with its slot values on Figure 88. Most of the editing as well as the commenting happens in the center pane. To display otherwise hidden comments in bright yellow and highlight those elements that are individual objects in MMS there are buttons on top. Each view also has a notification below its title about the number of comments inside and it can display when it was last edited, if the show elements feature is turned on. The export button in the center pane allows to create CSV tables and word documents through copy and paste. Yet, the word export as well as the direct print feature do not have a formatting on the same level as the PDF creation from an xml file. The direct conversion into a PDF in the View Editor requires additional commercial software, e.g. Prince [25] as used by JPL. Also, comments are not exported and there is the danger of shifted rows in printed and word tables as shown in Figure 84. Finally, on Figure 88 the right pane is opened, e.g. to show additional information about or to edit a selected element in the center pane.

E.4.2. View Editing

Editing views can happen directly in the left context pane, but most editing happens either in the central pane or in the right pane. To enable editing there is a button in the center, as displayed in Figure 88 and Figure 89. For the right pane there is a similar tap to edit only whatever element is currently selected, as being opened in Figure 89.

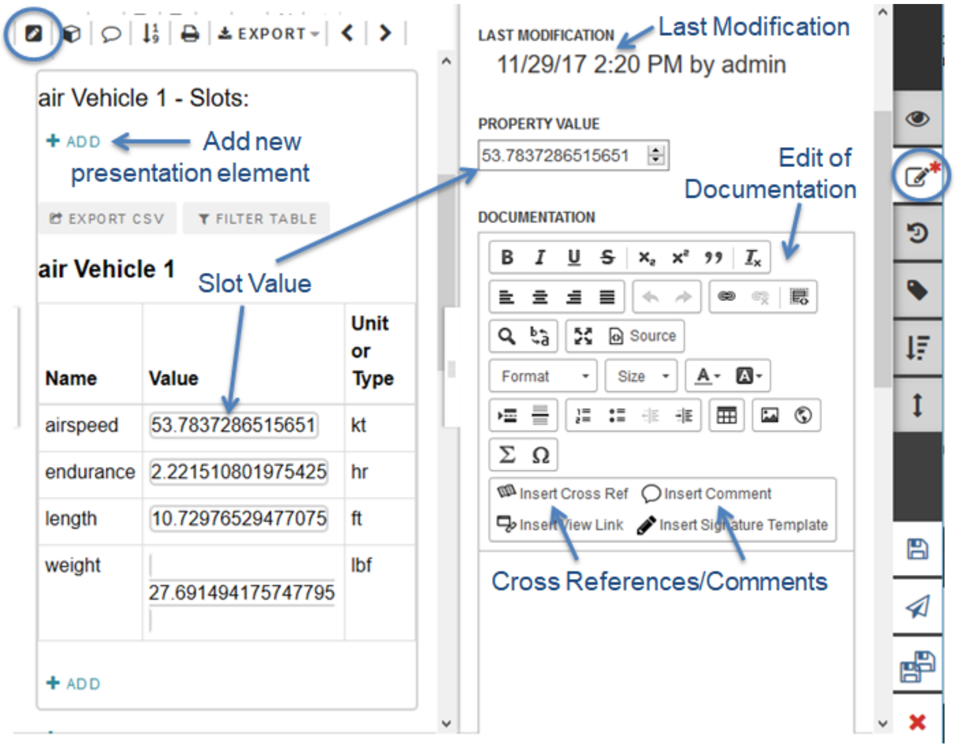
E.4.2.1. Editing View Elements

The views themselves can be added, deleted, renamed and their order and hierarchical position can be changed. When adding views from other projects, the single source of truth principle is maintained, i.e. it will be the very same view, e.g. including its comments. Deleting a view from another project does not delete the original view. If the user has limited rights for the original source project, its elements including views are not editable or not displayed, respectively. When reordering the views, it is also possible to change their position within the view hierarchy so that e.g. a view becomes a new subview or vice versa. “When a view that has subviews is selected to be moved, all of its subviews will move with it” [24]. Model conflicts arise, if the view hierarchy is changed by multiple users simultaneously in conflicting ways.

E.4.2.2. Editing Presentation Elements

Figure 89 shows the enabled editing of the center pane on the left and the right pane editing part for the slot element from the left table. The property value of the slot can be changed on the left side as well as on the right side. In both cases it would additionally cause a validation error by creating a blank name property for the slot element in MMS, as discussed earlier and displayed on Figure 86. Furthermore, the right editing window shows when and by whom the element was edited last and it displays a fully featured editing window for the element’s documentation.

Besides common text formatting features, including some for the insertion of tables, images, equations and hyperlinks, there are the buttons to insert a cross reference, a comment, a view link and a signature template. Cross references are links to the name, value or documentation of any other accessible model element in MMS. This not only includes comments created in the View Editor but also all other model elements from the opened projects and the projects it is using, even those that are not yet explicitly exposed in the View Editor. These cross references can improve consistency by e.g. including references to defined elements into a document and they can serve as an interface for directly changing those model elements, too. The view link is similar, but it links to views only, which can be directly opened by clicking on it. The created comments are hidden, unless the button in the center pane to show them is turned on. Comment elements stay intact, if their owning element is removed from the document, but not if it is deleted in MMS. Since comments can only be created through the enabled editing, one needs editing permissions to create comments. If this is not desired, a new branch with editing rights can be used of a read-only master model.



air Vehicle 1 - Slots:

air Vehicle 1

Name	Value	Unit or Type
airspeed	53.7837286515651	kt
endurance	2.221510801975425	hr
length	10.72976529477075	ft
weight	27.691494175747795	lbf

LAST MODIFICATION
11/29/17 2:20 PM by admin

PROPERTY VALUE
53.7837286515651

DOCUMENTATION

Figure 89: Editing a slot element in the View Editor

When editing text fields from MagicDraw in the View Editor they get added expressions to clarify their status as paragraphs in MMS <p> and </p>. Additional formatting and references are included analog in a text-based way, as mentioned in section E.2.1.7, where such expressions are used to define the formatting of the generated DocBook xml documents.

In general, names, vales and the documentation of elements can be changed in the View Editor. Changing other properties, e.g. the target element of an exposed relation in a table, does not work. This example would instead result in a changed name of the original target element. Also, most features exposed through OCL constraints cannot be edited.

Creating new model elements is only possible in very limited ways. Comments, for example, are their own elements and it is possible to create other class type elements e.g. through the cross reference button. Yet, those elements do not propagate properly into MagicDraw, since e.g. their position within the model is not specified. This is different for the created views from the View Editor, which do appear as part properties and are contained within their parent view or document element in MagicDraw. A workaround the limitation of not being able to create new model elements in the View Editor, is to create blank model elements in MagicDraw in advance and to expose them in the View Editor. Requirements without name or text can for instance be created in MagicDraw, to be detailed in the View Editor.

As seen on Figure 89 on the left, it is possible to add new presentation elements into existing views. They can be added everywhere in between existing presentation elements, no matter if created from a viewpoint or in the View Editor. Added presentation elements of deleted views stay intact, e.g. if the view is later re-added from MagicDraw. The presentation elements that can be added are text fields, tables, images, equations, comments and subsections. Those elements can then only be edited in the View Editor and do not get into the MagicDraw model. The text field is similar to the documentation field in Figure 89. The same is true for the table field, and it requires a new table to be created within it. The image field may not only include figures but also videos, using the source button to enter raw html with embedded iframe tags. An example using a youtube video is: “<p><iframe frameborder="0" height="390“ src="http://www.youtube.com/embed/52Vmukuf-Ks" title="YouTube video player" width="640"> </iframe></p>”. The equation field is for mathematical equations, using TeX [26]. Changing the order of the presentation elements of a view is possible via drag and drop in the right pane. The relative order of elements generated from DocGen viewpoints cannot be changed.

E.4.3. Branches, Tags and Additional Features

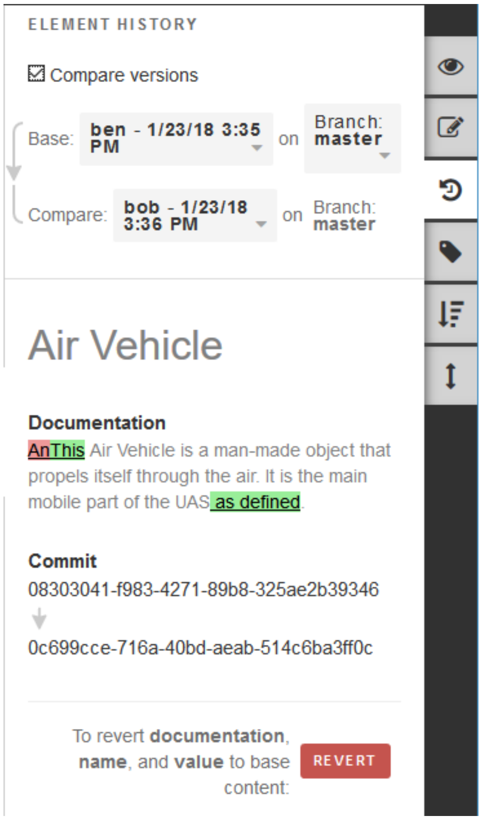
As indicated in the top right corner of Figure 88, the View Editor can create, delete and switch between different branches and tags of a project. Branches provide a separate workspace built by copying data at a specified time. Any additional changes do not affect the master branch. New View Editor branches include all added presentation elements and comments of the master branch. It is possible to synchronize branches with the same names in MagicDraw and MMS, if they were created first from MagicDraw and not in the View Editor. A merging also has to happen in MagicDraw. When deleting branches in the View Editor, they leave their data in Alfresco and they cannot be recreated again from MagicDraw.

Tags are similar to branches, except that they exist only in MMS and the View Editor and that they are read-only. They capture the state of the project in the View Editor as sets of permanently saved read-only data with a timestamp, to create "snapshots" of the View Editor at specified times, e.g. for reviews. When creating a tag for a certain point of time in the past, it does not include the diagrams as they were in the past. Also, there exists a bug that is supposed to get fixed with View Editor v. 3.3 and MMS v. 3.3 where a tag created from a branch incorrectly uses the images from the master branch.

The View Editor’s element history comparison feature is shown in Figure 90. It displays the changes of the documentation between two different commits to MMS. An option to revert those changes back is offered below. The state of elements from different branches or tags can be compared, too. If the element does not exist in one of the two investigated versions, it displays a notification. There is no document or view wide comparison feature. For this, one has to use e.g. MagicDraw to compare the model or export the document for a broader comparison.

To avoid data loss and model consistency conflicts through multiple concurrent users, the View Editor uses ActiveMQ [20] for live updates. These updates should inform the user when the current page that is being edited has been changed, when there is a conflict between two saved elements and when the element that is about to be edited is not up to date. However, with the current implementation this feature does not work as intended. Fixing this issue got postponed together with applying a potential workaround that uses forced repetitive messages to keep it running, and with MMS v. 3.3 ActiveMQ will no longer be required.

Another feature that is currently not working is the execution of jobs from the View Editor. It requires additional unreleased software called Platform for Model Analysis (PMA) (pronounce as Puma) by JPL as well as the open source automation server Jenkins [27]. It allows to execute and monitor scripts, whose execution is automated with Jenkins. Using a headless (i.e., with a GUI) MagicDraw with Cameo Simulation Toolkit jobs can be created to e.g. start simulations and re-generate views directly in the View Editor, as mentioned in section E.2.1.9.



ELEMENT HISTORY

☑ Compare versions

Base: **ben - 1/23/18 3:35 PM** on Branch: **master**

Compare: **bob - 1/23/18 3:36 PM** on Branch: **master**

Air Vehicle

Documentation

AnThis Air Vehicle is a man-made object that propels itself through the air. It is the main mobile part of the UAS as defined

Commit

08303041-f983-4271-89b8-325ae2b39346

0c699cce-716a-40bd-aeab-514c6ba3ff0c

To revert **documentation**, **name**, and **value** to base content: REVERT

Figure 90: View Editor element history comparison

Final lessons learned about the use of the View Editor from the surrogate pilot study are about using a blank MagicDraw model to completely create the content in the View Editor and about using a SysML model structure for the single purpose of generating a majorly text-based document in the View Editor. The first method is applied for issue tracking, by creating elements for issues while cross referencing them and by directly referencing comments in other model-based documents from used projects. These issues are entered into tables with their documentation for the issue description. Directly using set comments not only reduces the workload of creating a separate issue, but it also improves consistency, since the very same issue/comment is displayed in the issue tracking as in the commented document. Having MMS elements as issues also allows to cross reference between them. They do not need to exist within the MagicDraw SysML model, that only exists to provide an empty document and the project use of other models.

The second application is an approach we used to formalize a statement of work (SOW) and other project-related documents that use package structures in SysML to automatically create analogous sections and subsections that contain the documentation of the packages as paragraphs of text. This is a first step in an attempt to model traditional documents like a SOW. Additionally, contained diagrams are included as images and different types of elements are used to create different types of tables, e.g. for legal risk assessment or evaluation rating definitions.

Again, the SysML model as such is not in the focus. It is only a means to an end, i.e. to create easily shared and commented documents in conjunction with the more model-based documents, whose purpose it is to make normal SysML models more accessible.

E.5. Teamwork Cloud (TWC) and Cameo Collaborator

This final section about TWC [4] does not focus on its use in general. Instead the combined use of OpenMBEE with TWC is highlighted and NoMagic's TWC with the Cameo Collaborator [28] are compared towards OpenMBEE's MMS and View Editor. When using TWC together with MMS there does not exist a direct link between the two. Everything goes over MagicDraw with the MMS Sync of the MDK plugin.

TWC is a commercial server software to allow working together on the same MagicDraw projects through merging the concurrent work of all users. One opens TWC projects from the server with MagicDraw, locks elements for exclusive editing and re-commits the changes back to TWC. Such locking of model elements does not exist when working with MMS, where the locking mechanism of TWC fulfills the same role for working in MagicDraw and the live updates for working in the View Editor. To determine what editing rights a user has, TWC uses similar user roles as Alfresco. Hereby it is to note that their assignment is partially independent from OpenMBEE, e.g. it can be possible to have editing rights in the View Editor without even being able to open the model in MagicDraw. On the other hand, to commit a model to TWC from MagicDraw a successful login with editing rights in MMS is required for the coordinated sync from section E.3.1. In case the login to MMS fails and the MMS URL needs to be changed, one has to either wait for 10 minutes or restart MagicDraw before it is possible, because the URL is stored in the cache during that period. Saving a TWC project locally, i.e. not as a local copy, works without coordinated sync with its login to MMS and can be done without network access, too.

To be able to sync between branches in TWC and MMS, it is necessary to first create them in TWC and then commit them to MMS from MagicDraw, as mention before in section E.4.3. Merging branches also only works in MagicDraw. Added presentation elements from the View Editor are hereby not taken into consideration, since they do not progress over into the MagicDraw model. As such the branches in MMS do not merge directly but only through MagicDraw. The merging changes from MagicDraw are then committed to MMS. During the merge process it is possible to reject certain changes that should not (yet) be committed. Analog it is possible to pull changes from a master model into its branches.

Part of the merge process in MagicDraw is an in-depth model comparison, to determine which changes to reject or accept. From the two types of comparisons in MagicDraw, the Quick Diff and the Full Comparison, the later one requires two separate MMS logins to work. These comparisons in MagicDraw allow a broad-scale identification of differences between two versions of a model, while the View Editor element history handles only single elements at a time, yet including the presentation elements that are only in MMS.

To conclude with a comparison between MMS and TWC: While they have similarities, their main purpose is different. MMS is central hub for structured data to allow multi-tool, multi-repository, and multi-discipline integration. For this, MMS stores all model elements of a project, including their change history and view instances and tags for the View Editor in an open JSON format, to also allow data from other sources than MagicDraw. TWC's main purpose is the concurrent and distributed modeling in MagicDraw, including versioning and branches. For this TWC stores the Magicdraw projects, including their branches, model versions and element history.

Another comparison between OpenMBEE and the Magicdraw environment is made for the View Editor versus the Cameo Collaborator, which both appear very similar and fulfill identical roles. The View Editor requires model synchronization between Magicdraw and MMS in addition to the generation of view instances, to be able to directly access and edit the model elements stored in MMS. The Collaborator uses an easy one-way publishing of its exposed model elements, which are determined by analog view and viewpoint hierarchies or also by one of the provided predefined templates, e.g. for SysML or UPDM models. The Collaborator view instances are then stored as a file in Alfresco, where their access permissions are set and they can be opened or deleted. With the coming v. 19 of Magicdraw TWC will be able to hold the data for the Collaborator instead of Alfresco. This might allow a direct editing of the Magicdraw models stored in TWC in the future, too.

While the View Editor allows a direct editing of elements stored in MMS, the Collaborator can only be used to add comments to the exposed views, which cannot be brought back into MagicDraw. Also the Collaborator has no model element history comparison feature, lacking the access to the project history in TWC. The editing in the View Editor includes the creation of textual comments as well as the addition of presentation elements, which do not exist in the Collaborator. Yet, its commenting is more advanced compared to the View Editor, e.g. by allowing replies, resolving of comments and graphical comments on diagrams. The navigation within the views of the Collaborator as well as towards the model elements in the MagicDraw model is also more user friendly, e.g. with automatically created active hyperlinks on elements on exposed diagrams.

E.6. References

1. JPL. *Open Model Based Engineering Environment*. [cited 2018 04 25]; Available from: http://www.openmbee.org/.
2. OMG, *OMG Systems Modeling Language (OMG SysML)*, in *Version 1.5*. 2017.
3. NoMagic, Inc. *Magicdraw*. [cited 2018 04 25]; Available from: https://www.nomagic.com/products/magicdraw.
4. NoMagic, Inc. *Teamwork Cloud*. [cited 2018 04 25]; Available from: https://www.nomagic.com/products/teamwork-cloud.
5. NoMagic, Inc. *Cameo System Modeler*. [cited 2018 04 25]; Available from: https://www.nomagic.com/products/cameo-systems-modeler.
6. Blackburn, M., et al., *Transforming Systems Engineering through Model-Centric Engineering.* Stevens Institute of Technology, Systems Engineering Research Center, 2018. # SERC-2018-TR-103.
7. *MDK User's Guide.* JPL, 2016. # 12/19/16 2:31 PM.
8. Delp, C., et al., *Model Based Document and Report Generation for Systems Engineering*, in *IEEE*. 2013: Big Sky, MT, USA
9. *MDK DocGen User's Guide.* JPL, 2017. # 11/2/17 1:04 PM. Available from: https://github.com/Open-MBEE/mdk/tree/develop/src/main/dist/manual.
10. SyncRO Soft SRL. *<oXygen/>®XML Editor*. [cited 2018 06 20]; Available from: http://www.oxygenxml.com/.

11. OASIS DocBook Technical Committee. *The DocBook Schema Version 5.0*. [cited 2018 06 18]; Available from: https://docbook.org/specs/docbook-5.0-spec-os.html.
12. Tom Sawyer Software. *Tom Sawyer Perspectives*. [cited 2018 06 18]; Available from: https://www.tomsawyer.com/.
13. NoMagic, Inc. *Cameo Simulation Toolkit*. [cited 2018 06 19]; Available from: https://www.nomagic.com/product-addons/magicdraw-addons/cameo-simulation-toolkit.
14. OMG, *Object Constraint Language*, in *Version 2.0*. 2006.
15. *MDK Systems Reasoner User's Guide.* JPL, 2017. # 12/6/17 2:35 PM. Available from: https://github.com/Open-MBEE/mdk/blob/develop/src/main/dist/manual/MDK%20Systems%20Reasoner%20UserGuide.pdf.
16. Docker Inc. *Docker*. [cited 2018 06 12]; Available from: https://www.docker.com/what-docker.
17. The Apache Software Foundation. *Apache Tomcat*. [cited 2018 06]; 12]. Available from: http://tomcat.apache.org/.
18. Elasticsearch BV. *Elasticsearch*. [cited 2018 06 12]; Available from: https://www.elastic.co/products/elasticsearch.
19. The PostgreSQL Global Development Group. *PostgreSQL*. [cited 2018 06 12]; Available from: https://www.postgresql.org/.
20. The Apache Software Foundation. *Apache ActiveMQ*. [cited 2018 06 12]; Available from: http://activemq.apache.org/.
21. Alfresco Software, Inc. *Alfresco*. [cited 2018 06 14]; Available from: https://www.alfresco.com/.
22. Ecma International, *ECMA-404 The JSON Data Interchange Syntax*. 2017.
23. Alfresco Software, Inc. *Alfresco Documentation*. [cited 2018 06 14]; Available from: https://docs.alfresco.com/.
24. *MDK View Editor User's Guide.* JPL, 2017. # 11/20/17 9:26 AM. Available from: https://github.com/Open-MBEE/EMS-Webapp/tree/develop/Documents.
25. YesLogic Pty. Ltd. *Prince*. [cited 2018 06 14]; Available from: https://www.princexml.com/.
26. TeX Users Group. *TeX*. [cited 2018 06 15]; Available from: http://tug.org/.
27. SPI (Software in the Public Interest). *Jenkins*. [cited 2018 06 07]; Available from: https://jenkins.io/.
28. NoMagic, Inc. *Cameo Collaborator*. [cited 2018 06 15]; Available from: https://www.nomagic.com/products/cameo-collaborator-for-alfresco.

F. INTEGRATING SYSML, MBSEPAK AND MODELCENTER

Author: John Dzielski, Mark Blackburn

Implementing a Decision Framework in SysML Integrating MDAO Tools (Submitted to INCOSE Insight)

Abstract

This article describes an implementation of a decision framework modeled in SysML that can be executed with two different parametric analyzers. One of those analyzers provides the kind of cross-tool and cross-domain integration of simulation and analysis tools that will ultimately be required to implement model based design at large scales. The paper describes the decision framework and illustrates its implementation in SysML in the context of the design of a notional surveillance drone. The paper concludes with some observations about future directions and some of the difficulties that were encountered.

Keywords: SysML, decision framework, MDAO.

F.1. Introduction

Figure 1 illustrates a simplified perspective on a traditional systems engineering process (Blackburn 2018). The process is abstracted from a more detailed process described in (Cilli 2015). The process begins with a concept of operation (CONOP) phase that defines a need or gap to be filled by a system. The need is defined based on a business or mission analysis, or based on a set of stakeholder needs. The "What" part of the process involves defining the system-level requirements along with the objective measures and key performance parameters (KPPs) that will be used to evaluate a candidate solution. During the "How" part of the process, different system architectures are synthesized and sets of alternatives for each architecture are parameterized. During the "How well" phase, modeling, simulation, and analysis are used to analyze how effectively each alternative performs relative to the objective measures and KPPs. The role of a decision framework is to collect the objective measures and KPPs and present it to stakeholders in a way that allows them to determine which alternatives best suit their needs. In any real process, there will be multiple stakeholders who place different value or weight on each of the measures and KPPs, and it is important to be able to present to the decisionmakers the trade-offs that exist between them. This article describes a framework for doing this and a SysML implementation that uses a multi-dimensional design and optimization (MDAO) tool to implement it in the context of a notional surveillance drone problem.

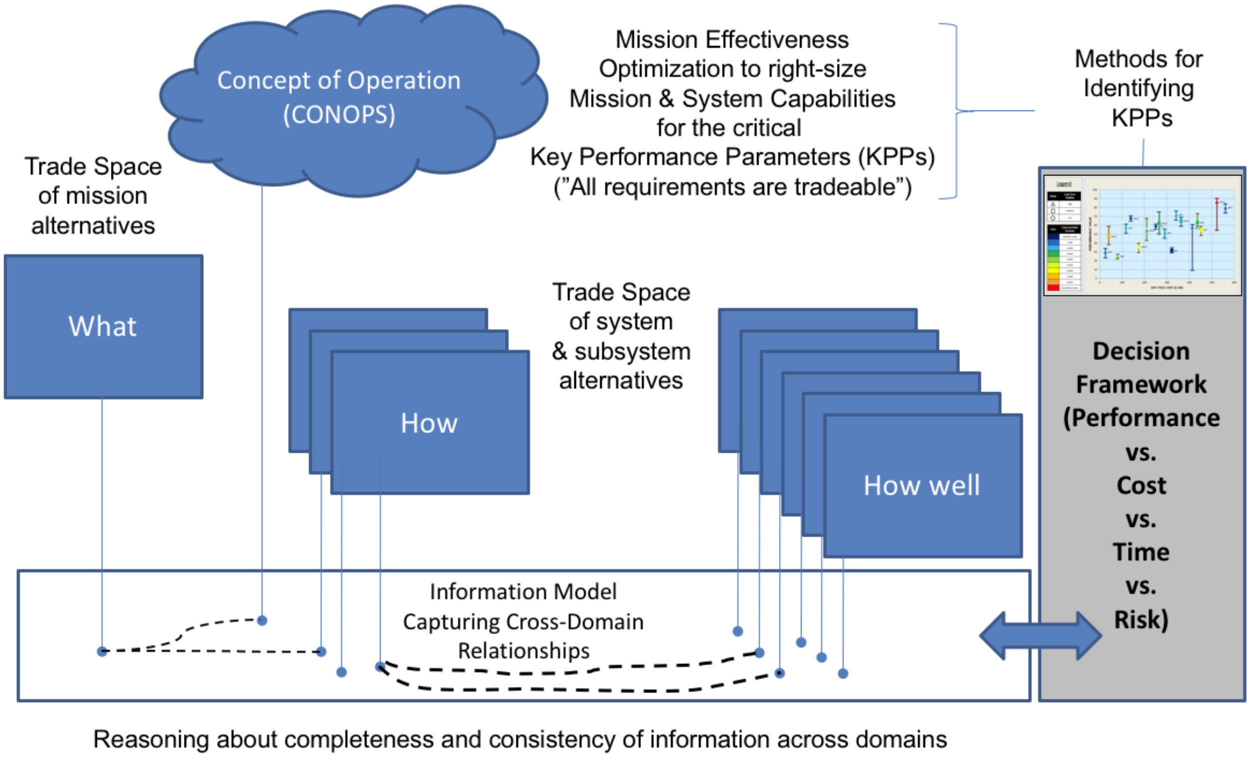


Figure 1. Simplified perspective of a systems engineering process (Blackburn 2018).

A successful transition from a document-based systems engineering process to a model-based process will require an ability to perform cross-domain and cross-tool analysis when evaluating the characteristics of a system. In the context of Figure 1, this is the process of determining the "How well" based on the "How." In a traditional process, the teams and tools used to perform these analyses are typically not linked digitally. Furthermore, the effort required to link these tools may be prohibitive and the linkages may ultimately be brittle if the integrator does not have control of the tools or their APIs. Finding efficient ways to link these analyses and capture the linkage in a digital model will be critical to enabling digital systems engineering processes.

ModelCenter® is a multi-disciplinary design and optimization (MDAO) platform that provides automation of cross-tool workflows in support of engineering analysis. The tool allows users to implement workflows that link analyses performed in a variety of widely used tools such as Matlab®, NASTRAN®, Ansys®, and SolidWorks®. This is only a partial list of tools that have been integrated, and the tool also allows integration of user-owned tools and workflows. ModelCenter® also provides integration with SysML through integration with MagicDraw® either through ModelCenter® or through a plugin to MagicDraw® called MBSEPak®. The implemented capabilities provide a means to automatically create complex workflows in ModelCenter® that are defined in the parametric diagrams of a SysML model, and to execute the workflow to perform cross-domain analyses and to execute trade studies.

The role of a decision framework is to relate the characteristics of a set of design alternatives selected by engineers to the value placed on those alternative by the stakeholders who are the owners or users and the decision makers. The output of the cross-domain analysis is a set of values for metrics and KPPs that are not directly comparable to one another. One reason that they are not directly comparable is simply because of differences in units. Another is that the numerical values may differ vastly in magnitude (e.g. milligrams vs. kilograms). The Integrated Systems Engineering Decision Management (ISEDM) Framework proposed in (Cilli 2015) provides a way to normalize these quantities in a way that expresses their value to a given stakeholder. Furthermore, different normalizations can be applied to reflect the needs of different stakeholders. Finally, data visualization tools can be used to understand the tradeoffs that exist within a set of alternatives and to guide the creation of new alternatives.

This article describes the results of an effort to implement a reference decision framework in SysML that performs the requisite analysis associated with a set of alternatives using an MDAO tool. The reference decision framework is described in the next section. In the following section, an implementation of the decision framework in SysML is described. To provide context, a simple example of a problem of designing a surveillance drone is introduced. To help understand the affect the choice of MDAO tool has on the process of building the SysML model, the framework was implemented to work simultaneously with a second tool Cameo Simulation Toolkit® (CST) from MagicDraw®. CST is marketed as a parametric solver for SysML diagrams. The final section discusses some conclusions drawn from using the tools and also discusses potential broader impacts associated with the framework.

F.2. Decision Framework

The Systems Engineering Body of Knowledge (SEBoK, 2017) identifies the development of objectives and measures as a critical part of a decision process. The objectives are the high-level concepts that give value to stakeholders such as performance, cost, and risk. For each objective, one or more measures are defined that quantitatively characterize the objective. Objectives and measures may be defined in hierarchies, and are often defined by a functional decomposition.

A second part of the process identified in (SEBoK, 2017) and discussed in the context of Figure 1 is the creation of alternatives. This process involves creating product architectures whose components provide the functionality to realize the objectives. A critical part of this process is identifying the key properties of an alternative and the characteristics that derive from those properties (Weber 2014)[7]. Here, properties are the attributes of a design that can be directly selected or influenced by the designer. Characteristics are those attributes that are indirectly influenced. For example, a designer can select a part's shape and what it is made of, but the part's weight results from those decisions.

In his thesis, (Cilli 2015) introduced the assessment flow diagram (AFD) as a tool for tracing the relationships between the properties of a system and the metrics and KPPs defined to measure its performance which will be referred to here as measures. Figure 2 shows an example AFD. The properties are identified with the "physical means" corresponding to the system architecture and the properties of its subsystems. At the top of the diagram are the list of measures and KPPs. In the figure, the "intermediate measures" are referred to as characteristics here. The AFD effectively describes a workflow for computing the measures and shows traceability between properties and measures.

---

[7] The use of the terms property and characteristics used here is reversed from that in the reference.

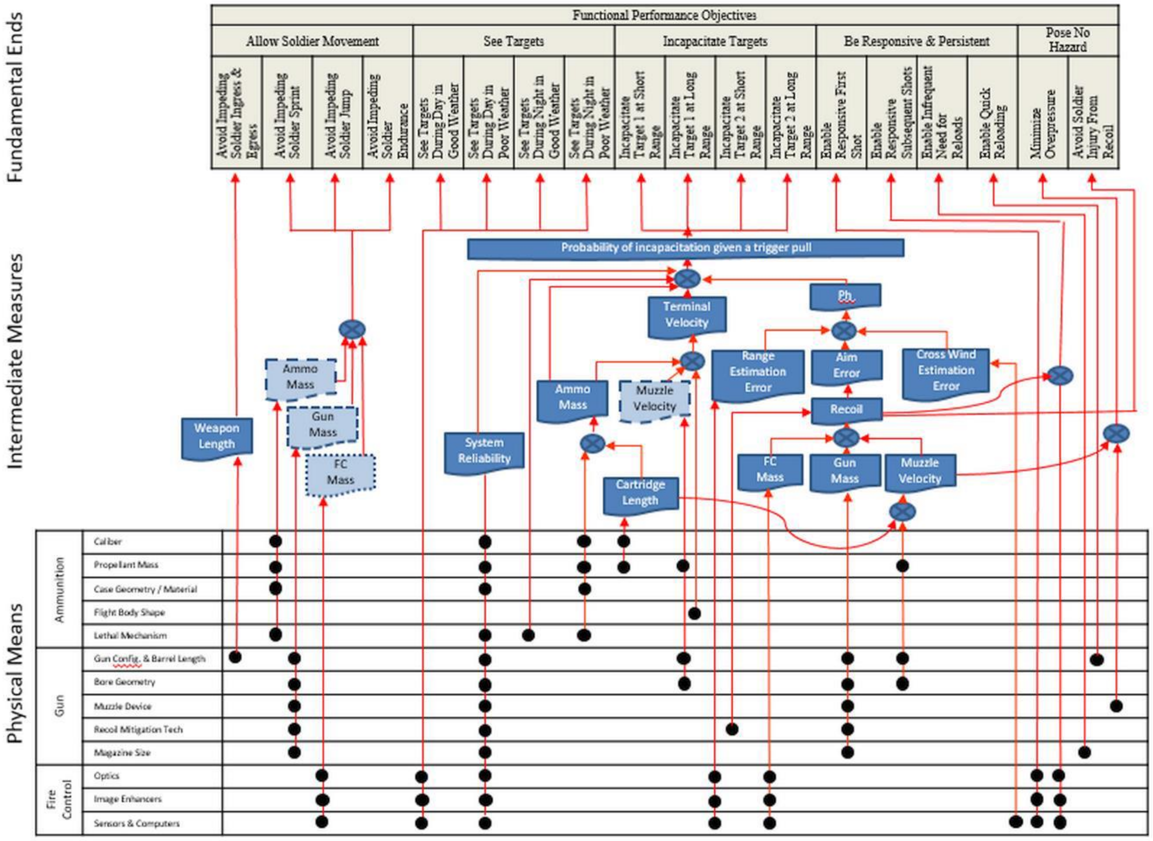


Figure 2. An example assessment flow diagram (Cilli, 2015).

(Cilli 2015) introduces the measure[8] scorecard as a tabular way of capturing and tracking the measures associated with alternatives in an analysis of alternatives or a trade study. The scorecard can be arranged in a spreadsheet with each row corresponding to an alternative and each column corresponding to one of the measures. Similarly, (Cilli 2015) introduces the value scorecard as a way of capturing and tracking the value applied by one or more stakeholders to the set of metrics associated with an alternative. For each entry in the measure scorecard there is a corresponding entry in the value scorecard. Each value is a monotonic function of the corresponding measure and maps the numerical value of the measure to a value scale of 1-100. An illustrative example of shapes can be found at (SEBoK, 2017), and one suggestion is that a value of 0 be associated with a measure that has not utility to a stakeholder and a value of 100 with a measure such that larger values provide no additional utility.

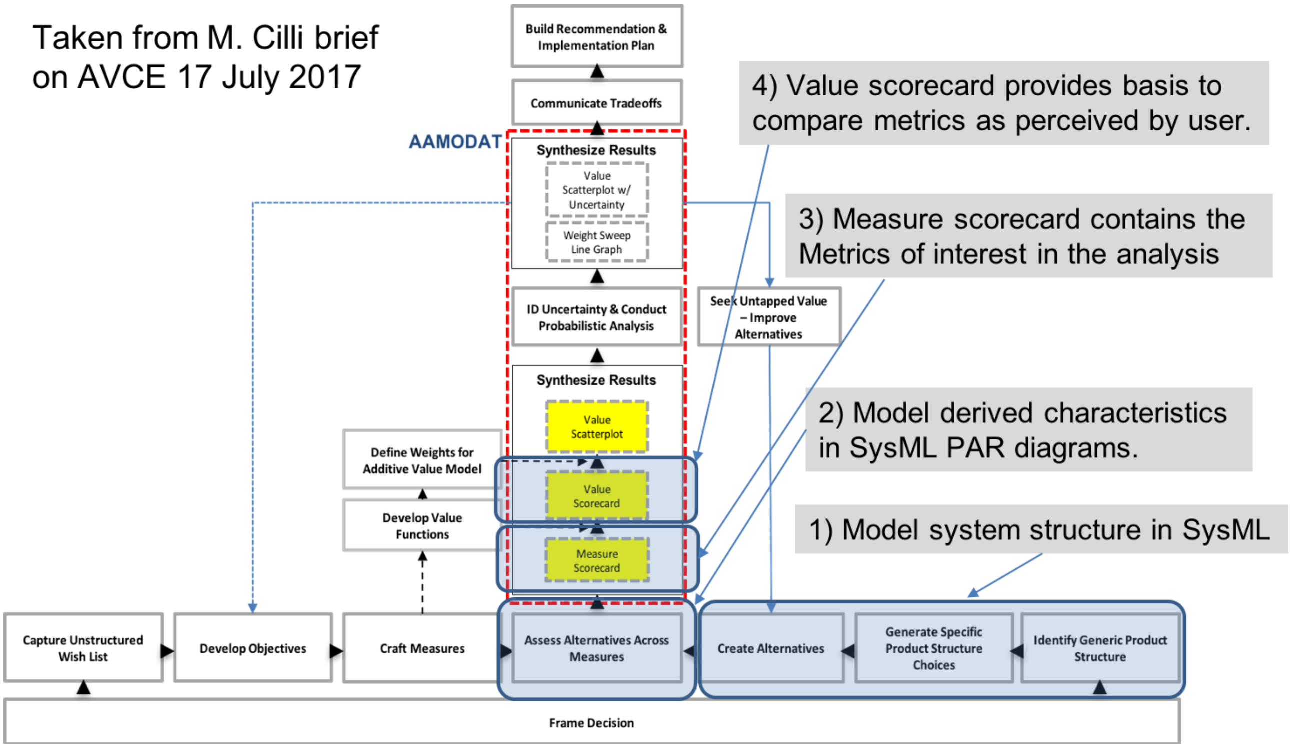
F.3. SysML Implementation of the Decision Framework

Figure 3 illustrates the ISEDM Framework described in the previous section. This section describes an implementation of that framework in SysML in the context of notional Unmanned Aerial System (UAS) surveillance drone. On the bottom left side of the diagram are steps associated with creating the metrics and KPPs. On the lower right are steps associated with

---

[8] The thesis uses the term "consequence" instead of the word measure used here. The latter is used here because it is believed to be more descriptive.

creating the alternatives. These steps are creating the generic and specific structures or architectures of the alternatives, and generating the alternatives themselves. This process identifies the properties that define the individual alternatives. The next step in the process is to model what in the AFD is called the intermediate measures and measures, and what are called characteristics here. This has been done with SysML parametric diagrams and constraints and produces results equivalent to the measure scorecard. Finally, an implementation for computing the values of alternatives is introduced.


Taken from M. Cilli brief on AVCE 17 July 2017

4) Value scorecard provides basis to compare metrics as perceived by user.

3) Measure scorecard contains the Metrics of interest in the analysis

2) Model derived characteristics in SysML PAR diagrams.

1) Model system structure in SysML

Figure 3. Annotated graphic illustrating a decision framework and steps implemented in SysML (used by permission)

The thesis (Cilli 2015) employed an example of an UAS for surveillance to illustrate the concepts, and that example has been adapted to demonstrate the application of the decision process here. The structure of a UAS is shown in Figure 4. The UAS consists of an Air Vehicle and a Payload. The Air Vehicle decomposes into an Air Frame with properties wingspan and altitude and an Engine with an engine that is either “Electric” or “Piston.”[9] The Payload decomposes into a pair of Imaging Sensors and a CommLink. The CommLink’s property is its weight and the Imaging Sensors properties are field-of-view, number of pixels, and pixel size. This set of values defines the properties that will make up each instance of an alternative.

---

[9] It would be natural to use an enumerated type here, but not all parametric analyzers that were used in this study worked with enumerated types.

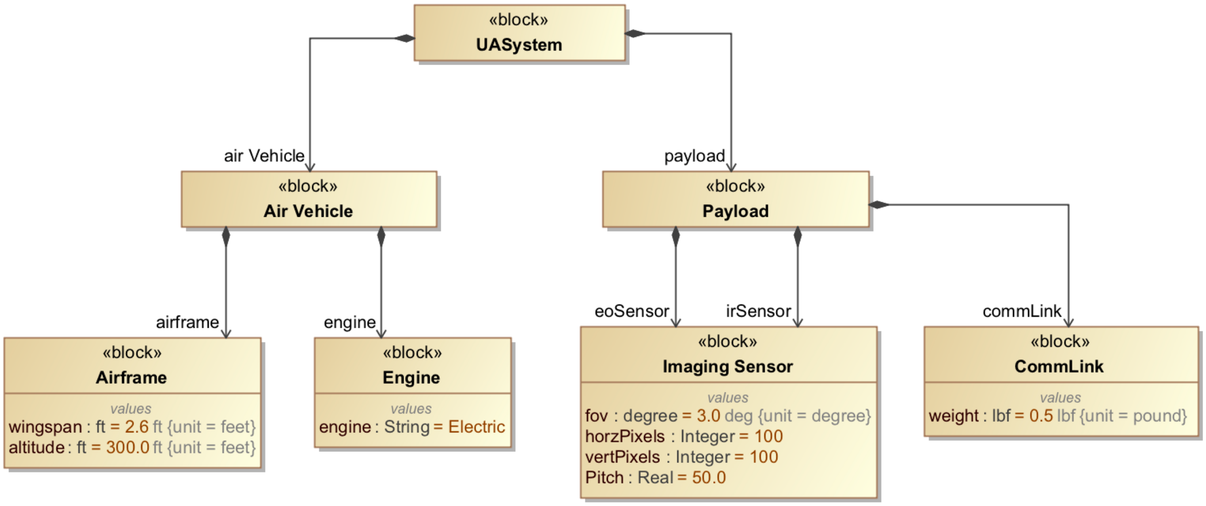


Figure 4. Structure of a UASystem showing the system properties.

The parts of a UAS along with their characteristics are shown in Figure 5. These are the intermediate measures in the AFD that are not necessarily directly of interest to the stakeholders as metrics or KPPs, but necessary to the calculation of those quantities. The names of the characteristics are largely self-explanatory and are indicated in the diagram as derived quantities. These quantities are calculated by defining constraint blocks in SysML and then binding the ports on the constraint blocks to parameters and other characteristics in parametric diagrams. The workflows required to compute the characteristics are created automatically in ModelCenter running as a stand-alone program or its MagicDraw plugin MBSEPak. Cameo Simulation Toolkit® provides a simulation capability to evaluate parametric diagrams. The measures associated with an alternative are the parameters that matter to the stakeholders or users of the system. A list of measures with descriptive names for the UAS are also shown in Figure 5.

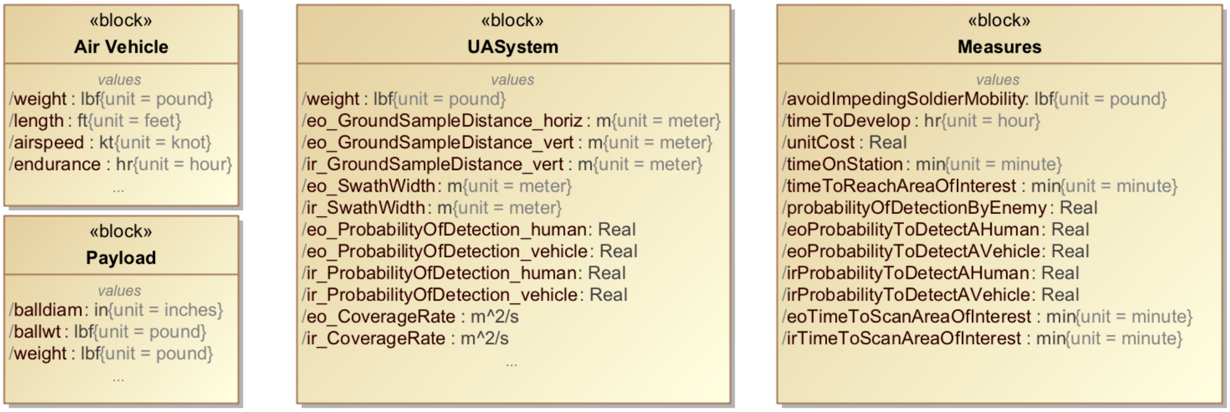


Figure 5. Blocks showing the characteristics and measures of the system derived from properties.

The constraint used to calculate values from measures is shown in Figure 6 along with two examples of generalized value blocks. A stakeholder selects a numerical value for each measure that is a walk-away value that is too small or large to be useful, a marginally acceptable value, a goal or target value, a value that would be highly desirable, and a value where larger or smaller values provide no additional usefulness. Default values for these points are 1, 10, 50, 90, 100, where a low value of 1 was selected to that ratios of values are always numerically valid. The two examples show a value function that decreases as the measure increases (weight) and increases with the measure (time on station).

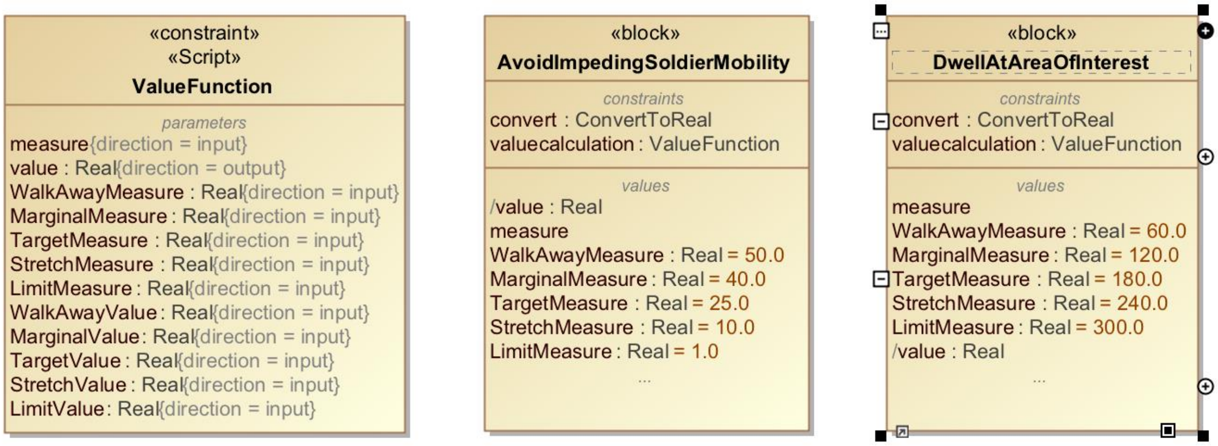


Figure 6. Constraint block defining the value function and two examples of value blocks associated with the UAS.

F.4. Defining a Trade-Study

A trade study can be built in SysML on top of the structures previously defined. An example trade study is shown in Figure 7. The study begins with the definition of a set of alternatives for the Payload and the Air Vehicle. These alternatives can be manually created as instances in the model, or read in from a formatted file or spreadsheet. An activity combines the instances of these alternatives in all possible pairings and creates an ordered list of UAS alternatives. A set of measures can be computed based on the alternatives and a set of values associated with those measure computed. How the measures and values are computed depends on the capabilities of the parametric analyzer or MDAO tool being used. The models built here used activities to apply the analyses. In the example below a trade study was implemented in Cameo Simulation Toolkit®, an activity sorted through the values for each alternative and eliminated the solutions that were not Pareto optimal. ModelCenter® has a number of built-in capabilities that support MDAO. Analysis of alternatives can be pursued there using built-in design of experiments capabilities or using functionality supporting numerical optimization. In either tool, alternatives can be automatically created as instances and saved in the model containtment tree.

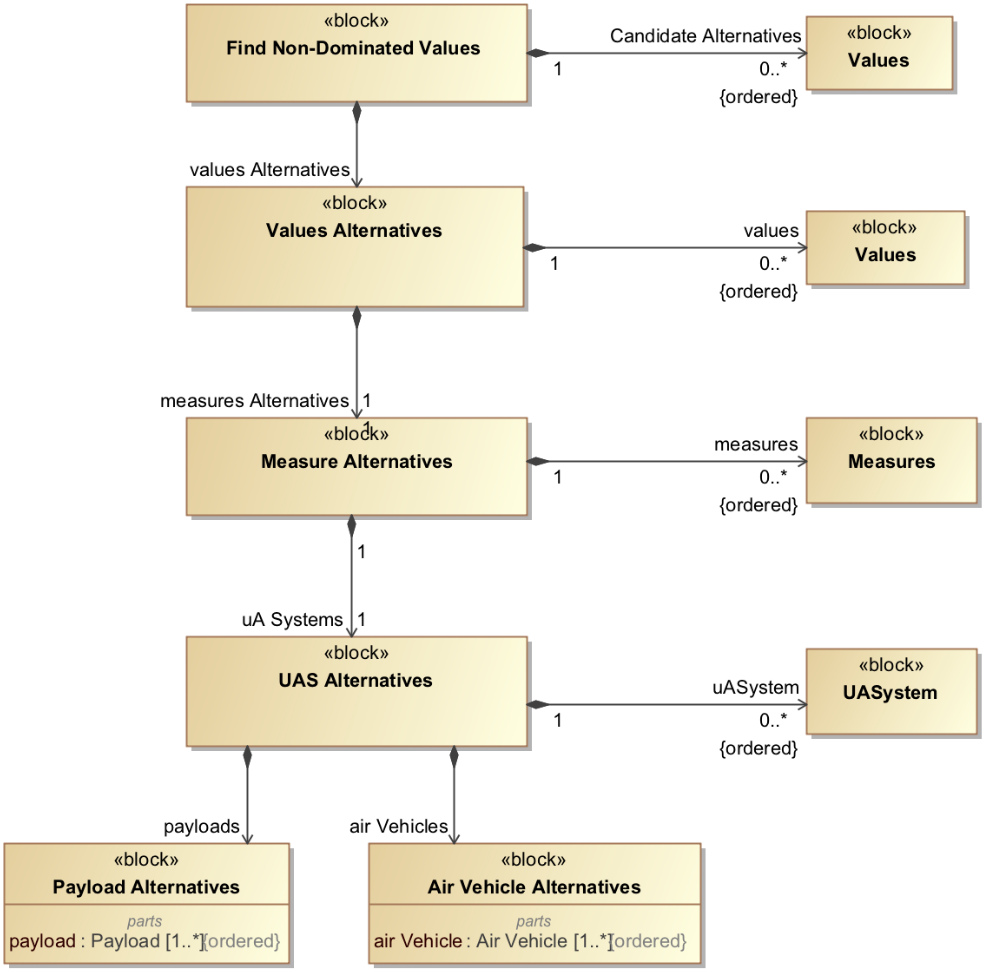


Figure 7. An example trade study built in SysML.

F.5. Summary and Conclusions

One of the goals underlying this effort was to show that it was possible to formalize a decision framework in SysML that could be implemented in a way that incorporated an underlying framework for cross-domain analysis. The ability to implement complex analyses currently done by independent teams and decoupled tools is critical to enabling a transition to model-based engineering. ModelCenter® is one tool that provides integration between tools used on large-scale engineering problems indicating the feasibility of linking model and analysis capabilities. Ultimately, a successful transition to model-based engineering will require capabilities that provide model integration that are tool agnostic. Sematic web technologies (ontologies) are being investigated as a possible means for addressing this need (Bone et al., 2018). The vision is that if analysis tools can have their interfaces described in a standard way, then tool integration can be handled automatically.

Another challenge that was identified as part of this effort has to do with what is standardized by the SysML standard. SysML defines a constraint as a relationship that must hold between a set of values bound to ports. The standard does not distinguish between values that an engineer might consider to be the inputs and outputs to a calculation. A consequence of this is that a set of parametric diagrams does not define a unique or unambiguous workflow for evaluating them. Also, tools may not create workflows for all valid SysML diagrams, and SysML diagrams that can be evaluated in one tool may not evaluate correctly in another.

F.6. Disclaimer

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 or SERC, 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.

F.7. References

Blackburn, M., 2018. *Transforming Systems Engineering Through Model-Centric Engineering*, Final Technical Report no. SERC-2017-TR10, Stevens Institute of Technology, Hoboken, NJ, August 2018.

Bone, M. A., Blackburn, M. R., Rhodes, D. H., Cohen, D. N., Guerrero, J. A., 2018. "Transforming Systems Engineering Through Digital Engineering." doi:10.1177/1548512917751873.

Cilli, M. 2015. "Improving Defense Acquisition Outcomes Using an Integrated Systems Engineering Decision Management (ISEDM) Approach." PhD diss., Stevens Institute of Technology (Hoboken, US-NJ).

SEBok, 2017. "Decision Management." http://www.sebokwiki.org/wiki/Decision_Management.

Weber, C. 2014. "Modeling Products and Product Development Based on Characteristics and Properties." In *An Anthology of Theories and Models of Design*, edited by A. Chakrabarti and L. Blessing, 327-352. London, GB, Springer-Verlag.

F.8. Rules to Follow for Using MBSEPak® and Cameo Simulation Toolkit®.

This is an extra section included in this report that was not included in the INCOSE Insight submission.

When building models with parametric diagrams that must be executed with MBSEPak® or Cameo Simulation Toolkit, there are certain rules that must be followed to ensure that the models execute with both tools. This section explains the rules that were identified during this project. These comments are valid for MagicDraw® version 18.5 SP3. Some issues are known to have been fixed in LTR 19.

Blocks should be related by directed composition relationships only. CST does not recognize blocks related by reference (aggregation or association). MBSEPak® will return an error about algebraic loops if relationships are undirected.

When building a model, diagrams should be tree structured. This means that from a parent block, there should be a single path following composition relationship to each child if that child is intended to represent a single model element. Both CST and MBSEPak will create an independent block and analysis flow for each path to a block.

Enumerated types are not fully supported in MBSEPak®. Problem appears when using instances. Use a string type instead.

Initialization of numerical arrays is not supported in MagicDraw®. This is fixed in LTR 19.0.

When creating instances in MagicDraw®, occasionally it will leave a slot value untyped or defined to be an opaque type. The model will simulate with CST but not in MBSEPak®.

When creating instances, parametric diagrams are not automatically executed. In fact, slot values may not be present at all. The MBSEPak® workflow requires that the user "Run" the model. This step will execute parametrics and assign values to slots. Using CST to simulate a block or instance will produce results in the simulation window but the results will not save back to the containment tree. The way to do this is to create a simulation configuration block that has the instance as its executionTarget and resultLocation.

G. MODEL BASED ENGINEERING: DESIGN AUTOMATION WITH SIMULATION SERVICES

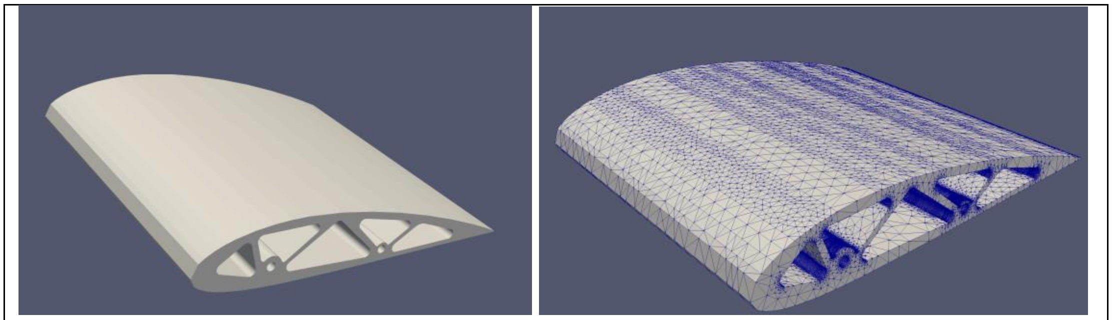
Authors: Kishore Pochiraju, Luigi Ballarinni

G.1. Design Automation with Simulation Services: Illustrative Example

We performed an illustrative example to demonstrate the generalized abstraction for embedding simulation tools. The workflow management from CAD to mesher to solver is shown with an airfoil example. This example paves the way for future work on simulation concurrency and pipelining for a more complex set of interacting simulations. The software tools chosen are agnostic to operating systems (most of them are open-source and require only standard programming language compilers) and computational hardware. The tools have an open API, suitable for automation and extension.

The meshing software chosen is Tetgen, which generates tetrahedra over arbitrary convex or concave hulls defined using surface meshes. Such an approach is ideal as most CAD software export high-quality surface meshes for 3D printers and this representation can be used for effective geometry transfer. The solver used, FreeFem++, has a unique modeling framework. Users can define arbitrary finite element spaces and PDES, which are solved transparently. This framework provides a rich description language for model specifications, and the solver abilities can be extended without the need for recompilation of the tool.

To demonstrate the simulation pipeline, we chose a typical airfoil and evaluated the aerodynamic lift and drag. The problem can be solved in both 2D and 3D. The 2D example is described here for the sake of brevity. The inherent symmetry in the geometry enables 2D solutions without the unwarranted simplification of the problem. Figure 91 shows the geometric representations, both as a solid body and as the meshed (tetrahedralized) representations.



(a) (b)

Figure 91: (a) Geometric Model of the airfoil as seen designed in the CAD Software. (b) Meshed (tetrahedralized) representation of the 3D geometry as produced by Tetgen.

The simulation task composition begins with producing a stereolithography representation (stl) from the CAD software. Stl is a surface representation, and the analyses typically require volumetric representations. The composer invokes a 3D meshing tool, TetGen from the simulation tool catalog. The tetgen invocation is performed with configuration switches. The switch (-p) indicates the polygonal surface data being provided in stl format, -k indicates to write the tetrahedralized mesh in vtk (Kitware, inc.) visualization format, and -q indicating that refinement of mesh to certain quality measure is necessary. The listing below shows the statistics of the geometry processing and the time taken to produce the meshed representation. The meshing produced 1,802,866 tetrahedra and 450881 mesh points within the volume. The process completed within 40 seconds of wall clock time.

```
$ ./tetgen -q  -k -p Wing.stl
Opening Wing.stl.
Delaunizing vertices...
Delaunay seconds:  0.381055
Creating surface mesh ...
Jettisoning redundant points.
Surface mesh seconds:  0.05236
Constrained Delaunay...
Constrained Delaunay seconds:  0.715957
Removing exterior tetrahedra ...
Exterior tets removal seconds:  0.064399
Refining mesh...
Refinement seconds:  32.5044
Optimizing mesh...
Optimization seconds:  0.662768

Writing Wing.1.node.
Writing Wing.1.ele.
Writing Wing.1.face.
Writing Wing.1.edge.
Writing Wing.1.smesh.
```

```
Writing Wing.1.vtk.
Output seconds:  4.08397
Total running seconds:  38.4724
Statistics:
  Input points: 63090
  Input facets: 21030
  Input segments: 11613
  Input holes: 0
  Input regions: 0

  Mesh points: 450881
  Mesh tetrahedra: 1802866
  Mesh faces: 3909300
  Mesh faces on facets: 607136
  Mesh edges on segments: 240459
  Steiner points inside domain: 147325
  Steiner points on facets:  64207
  Steiner points on segments:  228846
```

Next, the problem is communicated as a 2D surface external surface to the finite element solver. The solver produces lift and drag profiles around the aerofoil shape using the Joukowski method (for example, Modeling the Fluid Flow around Airfoils Using Conformal Mapping,http://evoq-eval.siam.org/Portals/0/Publications/SIURO/Vol1_Issue2/Modeling_the_Fluid_Flow.pdf?ver=2018-04-02-120817-147).

```
$ cat naca.edp

// 2D potential flow around a NACA0012 airfoil.
// Section for importing the mesh

 Th = loadtetmesh(“Wing.1”);
//Alternatively load vtk with iovtk.

// Define FE Space and seco$^{nd}$ order pressure interpolation.

 fespace Vh(Th,P2);     // P1 FE space
 Vh psi0,psi1,vh;              // unknown and test function.
 fespace ZVh(Zoom,P2);

// Define and solve the Joukowski problem.

 solve Joukowski0(psi0,vh) =     //  definition of  the problem
 int2d(Th)( dx(psi0)*dx(vh) + dy(psi0)*dy(vh) ) //  bilinear form
 + on(a,psi0=y-0.1*x)                      //  boundary condition form
 + on(upper,lower,psi0=0);
 plot(psi0);

 solve Joukowski1(psi1,vh) =     //  definition of  the problem
 int2d(Th)( dx(psi1)*dx(vh) + dy(psi1)*dy(vh) ) //  bilinear form
 + on(a,psi1=0)                     //  boundary condition form
 + on(upper,lower,psi1=1);

 plot(psi1);

 // continuity of pressure at trailing edge
 real beta = psi0(0.99,0.01)+psi0(0.99,-0.01);
 beta = -beta / (psi1(0.99,0.01)+ psi1(0.99,-0.01)-2);

 Vh psi = beta*psi1+psi0;
 plot(psi);
 ZVh Zpsi=psi;
 plot(Zpsi,bw=true);
 ZVh cp = -dx(psi)^2 - dy(psi)^2;
 plot(cp);
 ZVh Zcp=cp;
 plot(Zcp,nbiso=40,fill=1);
```

The listing above shows the control and configuration code provided to the FreeFEM++ tool by the task composer. This listing illustrates the template that needs to be used for aerodynamics analysis. The problem is solved by computing two fields (psi) and (psi1) with the two problems, posed in their variational form (Joukaowski0 and Joukowski1). These problems are formulated, and templates are validated by subject matter experts. The simulation task composer provides the mesh (geometry) and materials as appropriate (fluid density) and marks the appropriate boundaries. In this example, the boundary conditions are also standardized and are preset in the template.

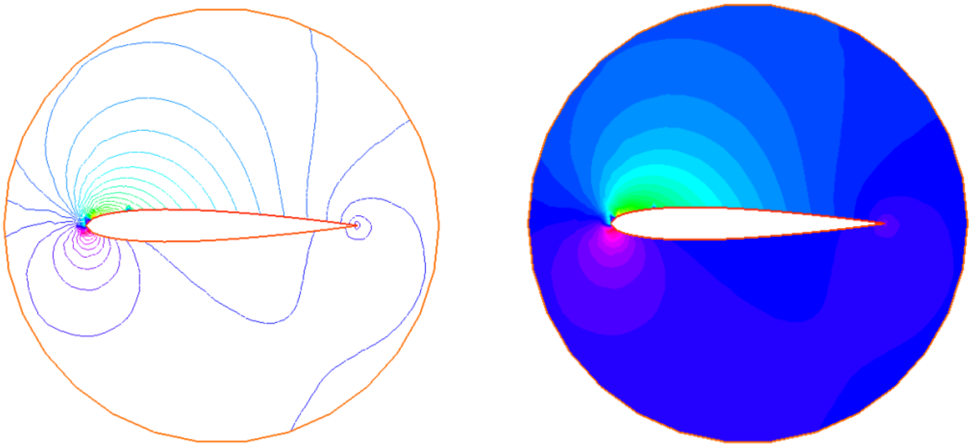


Figure 92: The pressure profiles around the airfoil shown both as line and filled contours around the airfoil. Blue indicates far-field pressure and red indicates the high-pressure regions.

Figure 92 shows the typical response outputs provided as fields from the solver. Post-processing template is necessary to compute and output response parameters such as the coefficients of drag (C_d) and coefficient of lift (C_l) for the airfoil shapes. Establishing accurate values for such parameters will be of utility for systems engineers and other design decision makers, who may need the behavior of the aerofoil suitably characterized and associated with decisions to be made in structure and propulsion domains.

H. INTEGRATION AND INTEROPERABILITY FRAMEWORK DECISION FRAMEWORK CASE STUDY

Authors: Mary Bone, Roger Blake, Benjamin Kruise, Tom Hagedorn, Ian Grosse, Chris Snyder, John Dzielski

H.1. Overview

The Interoperability and Integration Framework (IoIF), current state shown in Figure 93, is a high level semantic web technology (SWT) enabled architecture solution for the integration of data in the development and management of engineered systems. The IoIF is being developed using SWT to manage data interoperability of multiple digital artifacts.

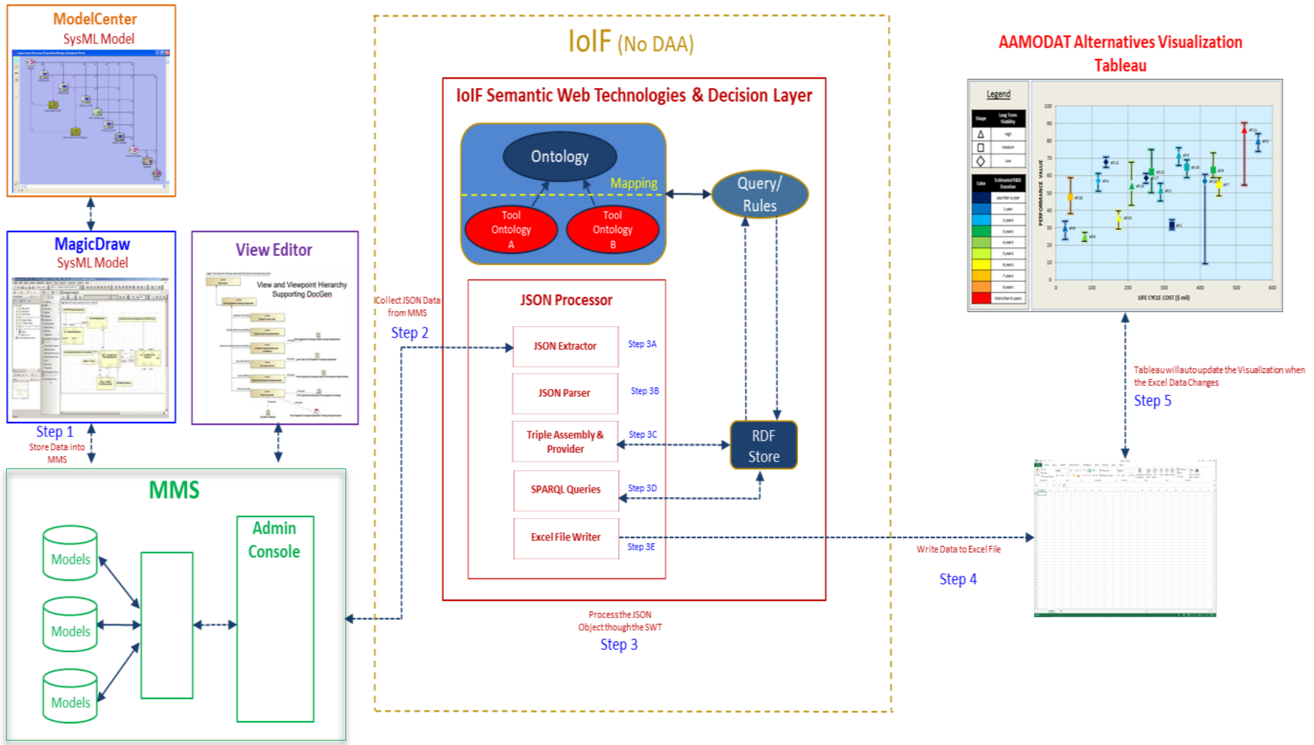


Figure 93 Current Interoperability and Integration Framework (IoIF) Architecture

One goal of the IoIF is to be able to retrieve data from any source and convert the data into a World Wide Web Consortium (W3C) linked data standard format such as JavaScript Object Notation (JSON) Linked Data (LD) or Resource Description Framework (RDF). Once the data is in a W3C linked data standard format then the data can be aligned with developed ontologies. After the aligning of the data with an ontology then queries, in SPARQL or another W3C query standard, can be developed and executed against the data. The queries can check for completeness and consistency of the data from multiple digital sources. The final goal of the IoIF is to notify users when data does not pass queries in regard to completeness and consistency, and compliance with an ontology. One example of a basic check would be for correct and consistent units for data. The IoIF could flag data and/or be given the authority to determine the digital artifact that has the authoritative source of truth (AST) for the specific data element and then the IoIF could use the AST data as the correct data to propagate to other artifacts. The IoIF has two main functions: 1) data acquisition and aggregation and 2) semantic query and reasoning which allows for consistency and completeness checking of the data.

H.2. Current State of IoIF and Demonstration

The demonstration of the IoIF presented in May of 2018, shown in Figure 93, was able to obtain data from the Model Management System (MMS) in JSON format, parse the data from JSON, convert the parsed JSON to RDF triples, execute a SPARQL query to return data needed for Excel table, convert the data needed for the Excel table from JSON to an Excel format and then export the Excel data. Once the data is exported in Excel it was able to be visualized via pre-defined Tableau scripts [187].

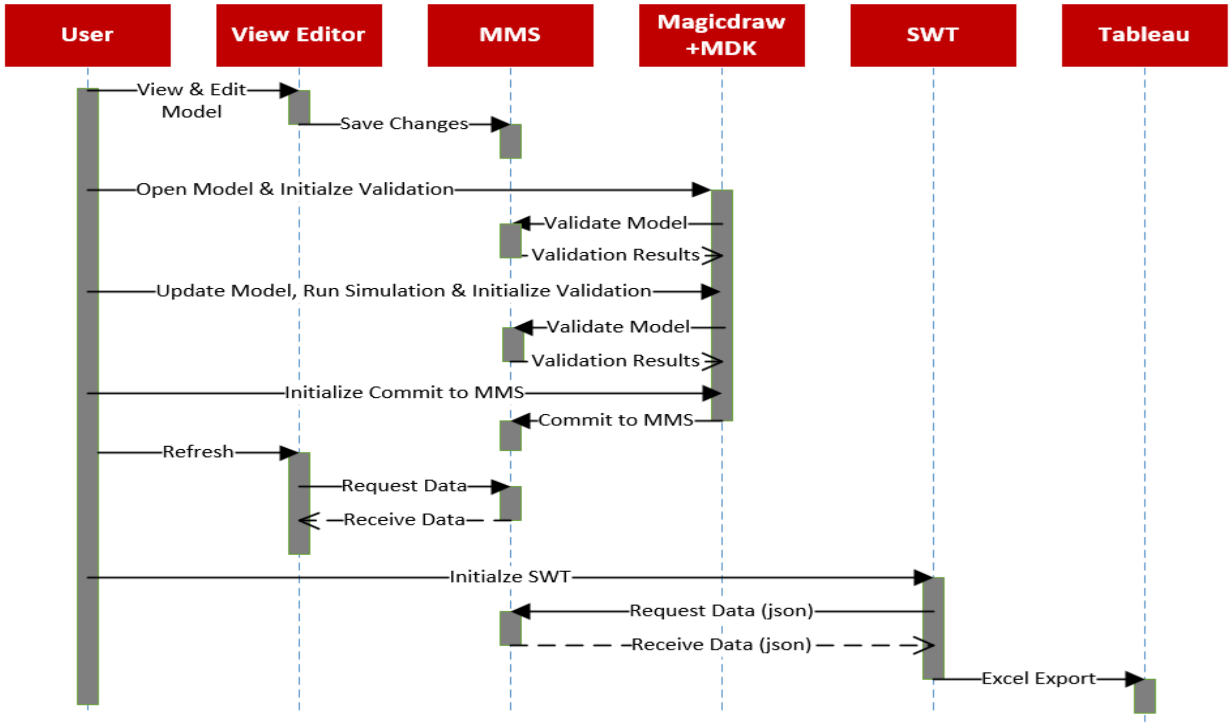


Figure 94 High Level Demonstration Sequence Diagram

Figure 93 shows the use case that was utilized to demonstrate the IoIF in its current state, as of May 2018, and Figure 94 shows the sequence of events between the digital artifacts shown in Figure 93. The current use case demonstrated that data required for Integrated Systems Engineering Decision Management (ISEDM) process, can be defined in a SysML model and then utilized across other digital artifacts via the IoIF. In this demonstration the data is specifically utilized by Excel which in turn is visualized in Tableau. In Figure 93 the demonstration data flows from the SysML tool on the left to top right. In future development the goal of the IoIF is to allow data to flow in any direction. For this demonstration Unmanned Aerial System (UAS) data was used.

The following sections will go into detail of the demonstration, Figure 93, which can be broken into three main areas: 1) SysML and OpenMBEE (View Editor & MMS), 2) IoIF, and 3) Excel and Tableau.

H.2.1. SysML and OpenMBEE Demonstration Details

The initial input for the demonstration involved capturing a UAS model in SysML (see Figure 95). This model allowed for multiple UAS alternatives to be generated from the UAS SysML model. Each instance of the UAS then had a unique set of parameters. The model, including all alternatives, were then able to be stored in the MMS of OpenMBEE using the Model Development Kit (MDK) plugin which exports the model data to the MMS, where it is accessible through its' Representational State Transfer (REST) Application Programming Interface (API). The MMS captures all model elements and is able to synchronize to the SysML model as edits/changes are made in the SysML model. The View Editor is also a feature of OpenMBEE that can be utilized to view the SysML model in a document like environment once it has been stored in MMS. Storing the model in MMS also allows the model elements to be edited in the View Editor and then synchronized with MMS and the SysML client modeling tool. While the main focus of the demonstration is the IoIF this part of the demonstration was able to show the capabilities of OpenMBEE and how it can be leveraged for data acquisition into IoIF.

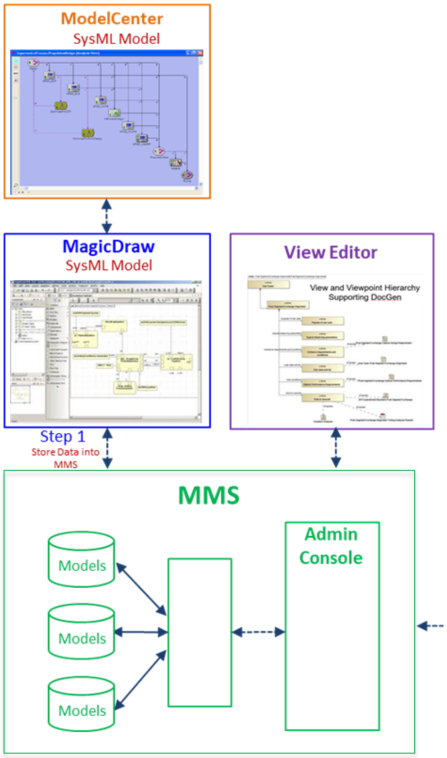


Figure 95 SysML and OpenMBEE

H.2.2. IoIF Demonstration Details

The main focus of the demonstration was the SWT capability of the IoIF architecture (see Figure 96). The current IoIF is a single layer with only the SWT and Decision layer without a Data Acquisition and Aggregation layer that is proposed for the final IoIF architecture. The demonstration was intended to provide proof of concept of the main technology being utilized in the IoIF architecture, emerging SWT. The first part of the IoIF demonstration sequence, shown in Figure 97, was to obtain the model data from the MMS. One of the benefits of using the MMS along with the SWT enabled IoIF is that the MMS exports data in a SWT friendly format, JSON. When the IoIF receives the JSON file it parses the JSON for the related model alternative data and stores the new JSON file. A SPARQL query is then used to find and fill in all the data elements needed for the Excel table. The table data in JSON is then converted to an Excel readable format and exported. The Excel table in this demonstration is specifically developed for UAS alternative decision-making data per the ISEDM process.

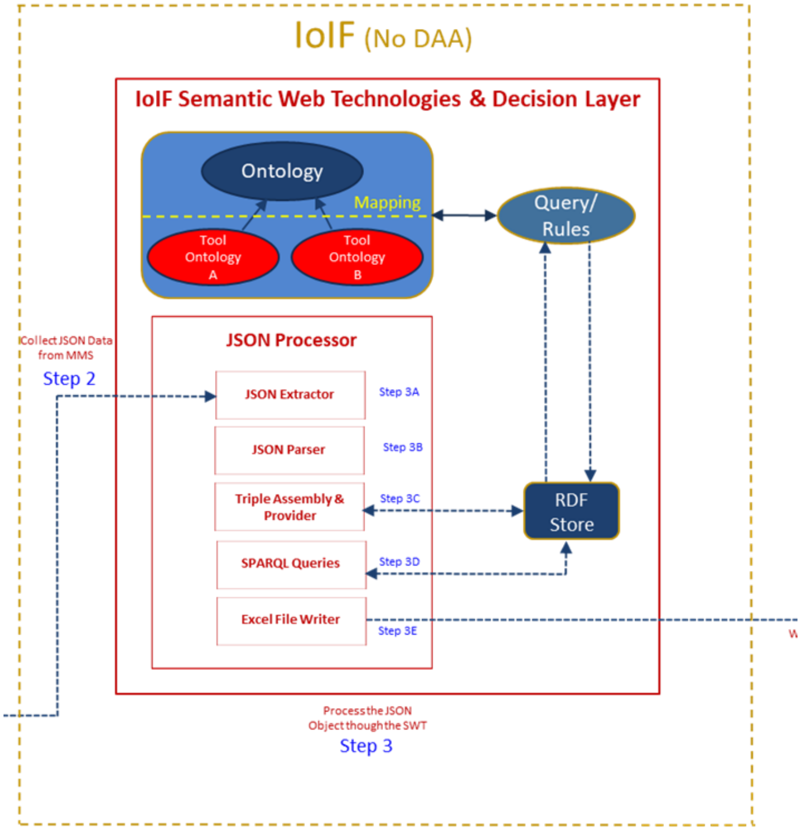


Figure 96 IoIF

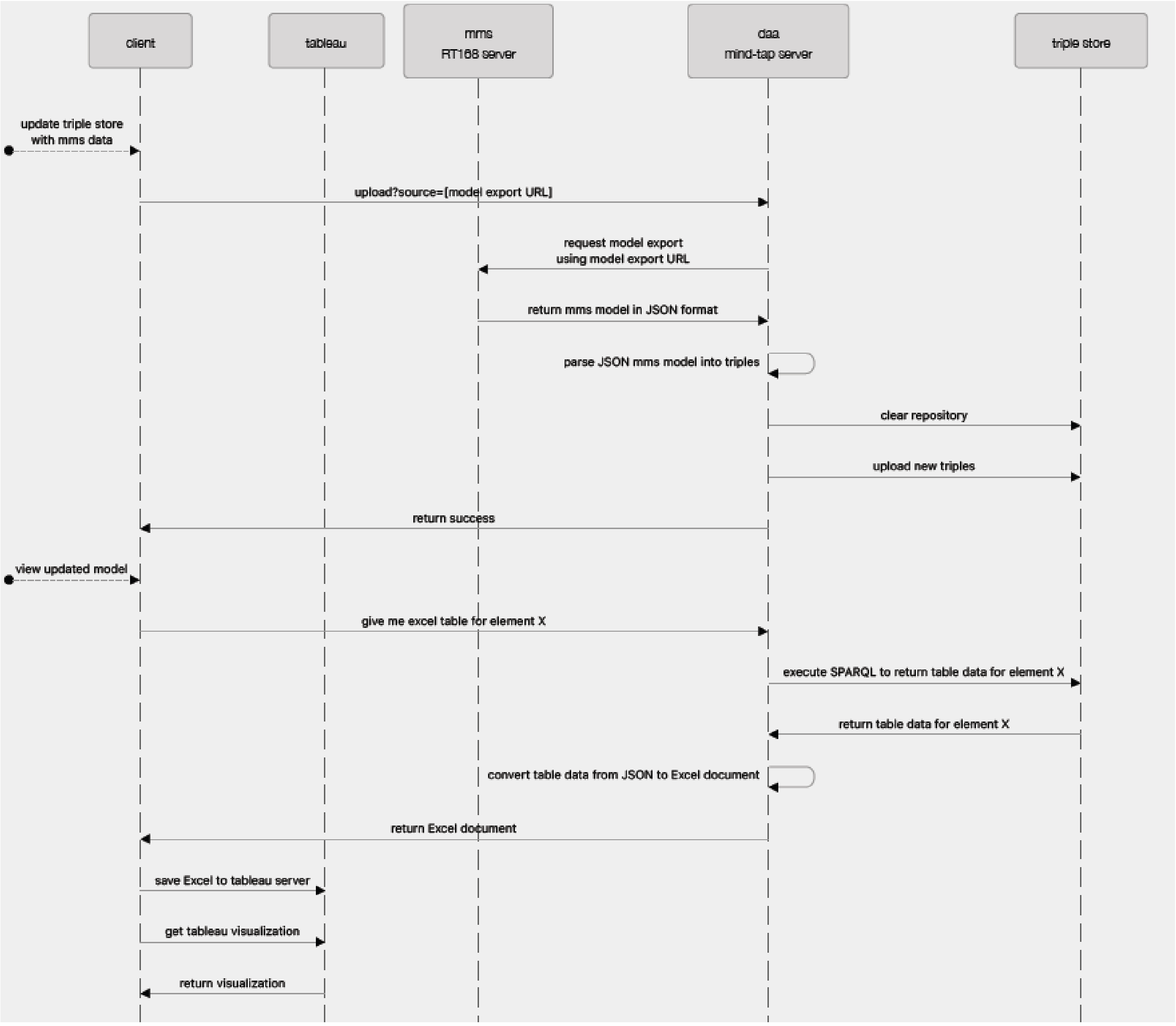


Figure 97 IoIF Demonstration Detailed Sequence Diagram

H.2.3. Excel and Tableau Demonstration Details

Once the data from the IoIF had been processed that produces an Excel readable file. The Excel file data was then utilized by Tableau to create pre-defined visualizations of the UAS alternative data (Figure 98). If the data in the SysML model is updated, the changes flow as characterized by the sequence (Figure 5) and the Tableau visualizations are updated. While the final goal would be to have the Tableau visualizations updated in real time there are limitations with Tableau that currently do not currently allow this, and a manual refresh is required in Tableau. In this demonstration the engine type of a set of UAS alternatives was changed from electric to piston in the SysML model and then the average performance versus unit cost in Tableau was updated to reflect this change via the IoIF, as shown in Figure 99. This demonstrated the ability of the IoIF to manage changes of data and make updates. A prior demonstration had shown the ability of the IoIF SWT to check, flag, and correct data units.

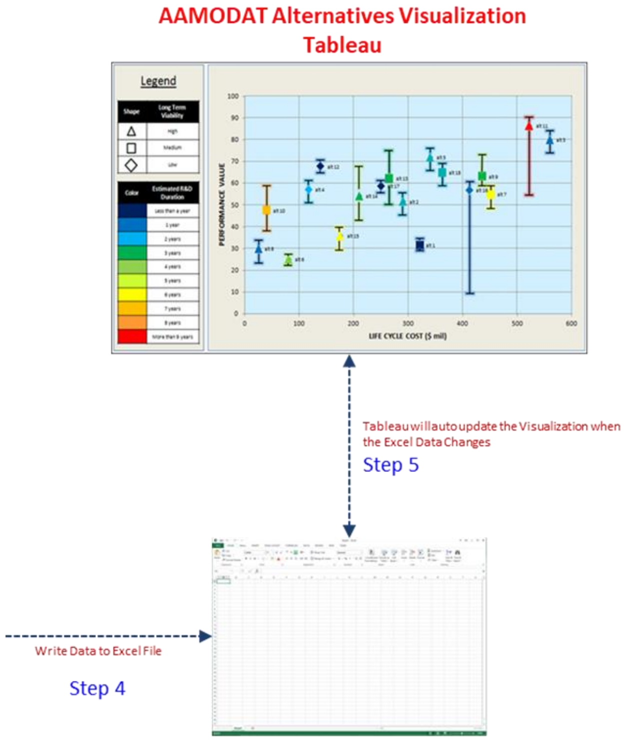


Figure 98 Excel and Tableau

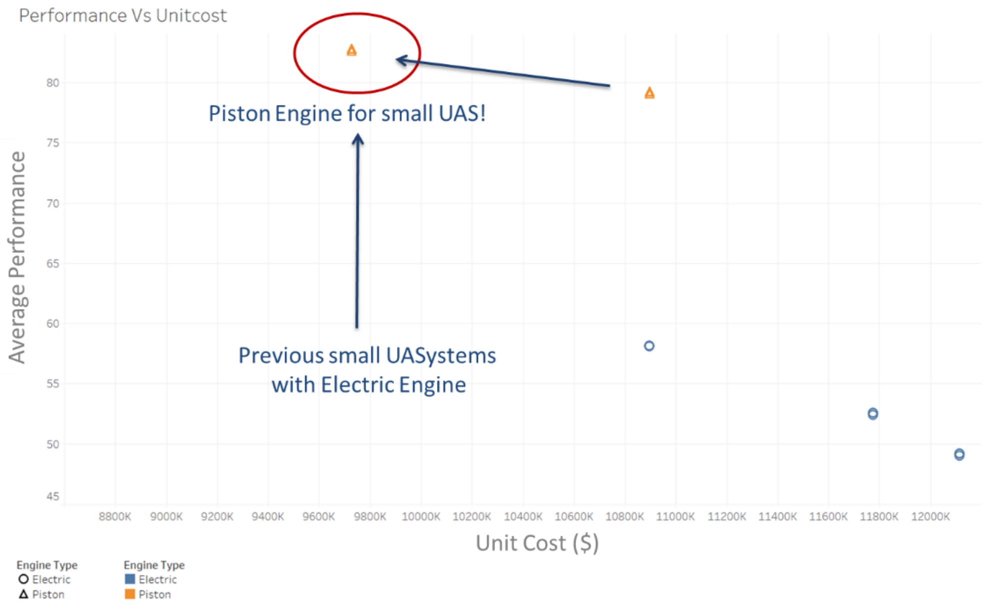


Figure 99 Average Performance vs Unit Cost ($) in Tableau

H.3. Future Demonstration of IoIF

The focus of the next demonstration of the IoIF will build on the current demonstration by taking the SysML data imported from the MMS and aligning it with ontologies as discussed Appendix B.

This is an important step because once data is aligned with an ontology then the full power of SWT can be applied to the data. Data can be queried and reasoned about for completeness and consistency. This also means that the limits of the queries and reasoning is based on the breadth and depth of the ontologies that are applicable to the data.

The next demonstration will also focus on starting to manage data inside the IoIF. While the MMS manages data externally to the IoIF, the IoIF will need to be capable of managing data since aggregation of data for all digital artifacts is one of goals of the IoIF. This will start developing how the IoIF can handle change management of data.

I. ASSESS ARMAMENT VIRTUAL COLLABORATORY ENVIRONMENT INTEGRATED MODEL BASED ENVIRONMENT

Author: Rick Dove

I.1. Context

First the task description will set the overall context, followed by some background on what the task description refers to as a three-system perspective.

The task description appeared in the A013 Interim Technical Report SERC-2017-TR-110, Update: July 15, 2017, by Mark Blackburn, as follows:

> [Task] 11. We were asked to provide a more detailed analysis of the Armament Virtual Collaboratory Environment (AVCE) integrated Model Based Environment (iMBE requirements. We initially looked at the requirements, but in attempt to do the analysis started to identify additional use cases not reflected in the model as shown in Figure 11. ARDEC then did deliver the AVCE iMBE model, and we developed a set of View and Viewpoints for the model to allow for us of MDK/DocGen. While the model is well structure, the View and Viewpoints modeling process revealed some minor inconsistencies, which we shared with ARDEC. While ARDEC has finished the Systems Requirement Review (SRR) for AVCE iMBE. Rick Dove joined the RT-168 research team.
>
> - Rick has done some interesting work [in] INCOSE’s Agile Systems Engineering Life Cycle Model (ASELCM) project, and specifically in terms characterized by the ASELCM Pattern of Three Concurrent Systems. Rick will use this context to look at the AVCE iMBE model from this three-system perspective

The three-system model arose from the INCOSE project (led by this author) that set out to “discover” an Agile Systems Engineering Life Cycle Model (ASELCM) by analyzing successful mixed discipline agile systems engineering processes for common life cycle operational attributes. The three-system model is called the Agile Systems Engineering Pattern (outlined below), and is the work of Bill Schindel, a co-lead on the ASELCM project. The context of Task 11 work and this summary is rooted in the operational aspects agile systems engineering, as both the RT168 project and ARDEC recognize the need for agility in the iMBE process and the products that iMBE models.

The definition of agile systems engineering is rooted in what it does, not how it does it. What it does is respond effectively in a life cycle environment that is capricious, uncertain, risky, variable, and evolving. How it does that is a product of analyzing response requirements dictated by the nature of the life cycle environment. The design and evolution of an operationally effective agile systems engineering process is itself a systems engineering activity, one that requires an attentive emphasis on problem space characterization and ongoing evolution. This summary will cover methods for developing and maintaining problem space characterization, and identifying and tracing the life cycle response requirements dictated by that characterization. If you don't know where you are going, any road will do.

The principal objective of Task 11 is to identify operational requirements that an iMBE tool and its supporting infrastructure should address and support. An important related objective is to identify a value proposition that can encourage mixed-discipline engineering teams and their management to appreciate and embrace a model-based systems-engineering approach, the lack of which appears to be impeding "customer" interest at ARDEC.

I.2. Three System Model – The Agile Systems Engineering Life Cycle Model Pattern

Agile systems engineering encompasses three nested concurrent systems, depicted in Figure 2 as an iconic pattern (Schindel 2016).

The ASELCM Pattern establishes a set of system reference boundaries. Whether the systems of interest are small or large, human or inanimate, flying through the air or performing business processes.

This ASELCM Pattern particularly refers to three major system reference boundaries, and within those, six subsystem reference boundaries. These are all logical boundaries (defined by the behavior, not the identity, of systems), and are depicted by the iconic diagram in Figure 1.

- System 1: The Target System, the subject of innovation over managed life cycles of development, deployment, and support.
- System 2: The Target System Life Cycle Domain System, including the entire external environment of the Target System—everything with which it directly interacts, particularly its operational environment and all systems that manage the life cycle of the Target System. This includes the external environment of the operational target system(s), as well as all the (agile or other) development, production, deployment, support, security, accounting, performance, and configuration management systems that manage System 1.
- System 3: The System of Innovation, which includes System 1 and 2 along with the systems managing (improving, deploying, supporting) the life cycle of System 2. This includes the systems that define, observe, analyze (as in agile software process retrospective), improve and support processes of development, deployment, service, or other managers of System 1.

System 1 is contained in System 2, which is contained in System 3. All are (or at least should be) happening simultaneously, effectively an organic complex system motivated by self-preservation to evolve suitably in an uncontrolled operational environment. Think of the arrow-pointed pipes of Figure 1 ideally as a circulatory system – not as channels for intermittent communication; but rather as pipes with constantly circulating information fluid. This circulatory system brings nourishing information and also provides regulation – policing the effectiveness, removing the dysfunctional, correcting the crippled.

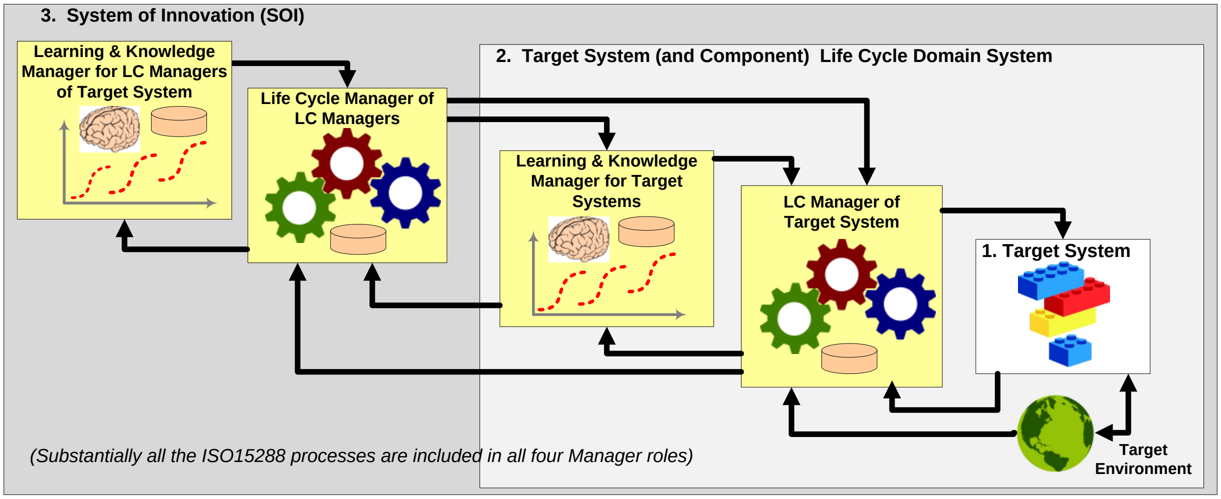


Fig. 1. Iconic view of the ASELCM Pattern reference boundaries (Schindel, Dove 2016).

I.3. Research

The research method began with a search for operational experiences in mixed discipline modeling and simulation environments where a face-to-face interview could occur (without incurring travel funds). Only one was found, the Common Engineering Environment (CEE) at Sandia National Labs (this researcher is within a 3-hour drive of Sandia), which appeared promising; though CEE is focused on simulation and analysis tools other than common Model Based Systems Engineering tools. The initial request for a half-day interview with personnel responsible for the support and service of the capabilities was positive but failed in securing and actual interview for what appeared to be security issues. Knowledge of the operation was limited to publicly available materials.

From (Goebel and Pavlakos. 2012), after 2+ years of CEE existence:

> The Common Engineering Environment (CEE) is a set of Preferred Engineering Software, Infrastructure, and Support Services that enable a Highly Functional Engineering Environment with seamless and ubiquitous access to Computational Engineering Capabilities.
>
> CEE Goals:
>
> - Facilitate the identification and use of a common, preferred set of tools and infrastructure for computational engineering.
> - Provide consistent and easy-to-use methods for customers to access, and use software tools.
> - Provide consistent support for engineering tools and infrastructure in the preferred set.

Later in (Pavlakos 2014) we get a more detailed view:

Sandia's vision for the CEE is a set of preferred engineering best practices, processes, training, tools and software, support services, and shared architecture that enables an integrated engineering environment with easy access to tools and capabilities, lab-wide efficiencies, and a disciplined approach to engineering."

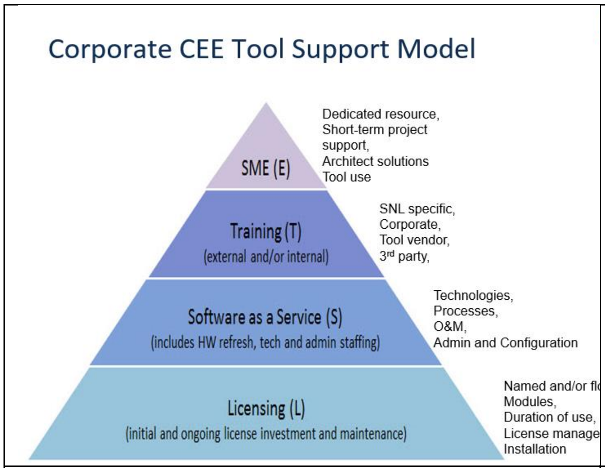


Fig. 2. Here we see the emphasis on a tool support infrastructure for engineers that will use tools licensed by CEE.

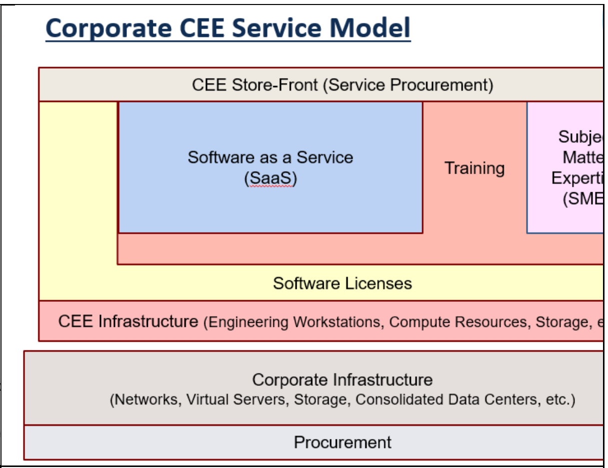


Fig. 3. Here we see the emphasis on a service infrastructure that provisions and provides tool capabilities for engineers

My take away is the need for an organizational support and service capability to enable engineers to focus on their development efforts rather than tool acquisition, provisioning, tool access, training acquisition, and administrative efforts. The iMBE capability cannot be supplied as an off-the-shelf product from a single-source product supplier, as it necessarily requires access to and integration with tools that may be project, program, or organizational specific. This implies either a need for an internal iMBE service group or an equivalent contracted service capability.

I.4. Literature Reviews of Operational Infusion Experiences and Plans

Research then moved to a literature search. The issues of transformation and infusion have bearing on the operational requirements as full-scale cutover to an iMBE-like environment is neither recommended in the literature nor practical. Many sources (see Bibliography) were found and read for potential influence on this task; but generally little had direct and relevant bearing on establishing requirements for an operational environment beyond the work found and done at NASA/JPL on a Concept of Operations for an Integrated Model Centric Engineering environment (Bayer et al. 2011, Bayer et al. 2012, Lin 2014); Ron Carson related experience at infusing MBSE into Boeing product development with a presentation *(*Sheeley, Malone, Palmer, Carson 2014 repeated in 2018); a Nuclear Plant case study paper of MBSE application by Thales (Navas, Tannery, Bonnet, Voirin 2018), and a Federal inter-agency compendium of infusion challenges and recommended mitigations (MBSE Infusion Task Team 2017).

I.4.1. NASA/JPL

An Integrated Model Centric Engineering (IMCE) Concept of Operations document (Bayer et al. 2011) focused on areas of "largest beneficial impact on project development and operations, as well as those that are needed for early infrastructure development." The document was intended for two distinct audiences: The IMCE Architecting team, and potential adopters, users and funders of IMCE capabilities. Goals and objectives for the ConOps were articulated in the document as follows:

> Goal 1: The IMCE Team understands and is able to articulate to project users why project users should, and how they can, use IMCE to accomplish a model-centric project development.
>
> Objective 1A: Collect, synthesize and document significant problem areas in the current flight project development environment, and how IMCE addresses them. (This is now documented separately in an internal memorandum [3]).
>
> Objective 1B: Describe, through development of selected scenarios, how people supporting a JPL flight project would interact with models, model-centric tools, and each other throughout the project life cycle.
>
> Objective 1C: Discover and document the characteristics of a flight project lifecycle as executed in a model-centric environment.
>
> Goal 2: The IMCE Team has valid use cases upon which to base the IMCE Architecture.
>
> Objective 2A: Deliver the Concept of Operations with form and content appropriate for the IMCE Architecting work and the Architecture Description Document.

My takeaway: JPL architected, modeled, and developed a Concept of Operations for their product development “enterprise” as a precursor for development of an IMCE capability. This helped them identify operational requirements before designing a solution. Notably, they collected problem areas in the before-IMCE development approach and showed how IMCE would address them (internal document) – which amounts to an evidence-based value proposition for users.

A subsequent paper (Bayer et al. 2012) discussed “Early Formulation Model-Centric Engineering” and provided 12 lessons learned. This paper addresses issues and lessons learned in the transformation to IMCE that inform our interest in the nature of the operational environment and should also be useful to ARDEC early iMBE introduction planning. Selected relevant briefs from 8 of the 12 lessons-learned follow:

1. **Unity of Leadership is Essential.** In the first infusions, management support for the effort must to be clear and consistent (verbatim from the paper).
2. **Early Efforts Draw on a Limited Pool of Talent.** Similarly, to the “Unity of Leadership” above, the first infusions will not have the benefit of an engineering pool with ubiquitous modeling skills (verbatim from the paper).
3. **Leverage Learning with Synergistic Work.** With a limited pool of modeling talent and three projects in need of modeling experts they allowed the experts to participate in all three simultaneously, which they believed outweighed the lack of full time commitment on a single project. “We have found this belief to be fully validated: the learning that has been shared between the three efforts has been enormously beneficial for all, and has clearly accelerated the institutional infusion.
4. **Team Organization Matters.** While we have found that descriptive modeling can be done by almost anyone with the basic training, the additional rigor and consistency needed for

quantitative analysis requires us to designate a smaller team of people who are modeling experts and who can apply best practice to the official configuration managed project models. Presently we have a core modeling team of a half dozen or so, within a larger team of 20 or so engineers. The experienced systems engineers provided guidance to keep the modeling focused on providing useful information, as well as mentoring of the core modelers who tended to be more junior. Frequent (daily) interactions were crucial to getting useful products (verbatim from the paper).

5. **Everyone Needs Training, but to Different Levels.** In the usage model above it is clear that all three groups need to receive training commensurate with their level of interaction with the models. Different levels of modeling familiarity are required, thus resulting in different levels of training. Working with IMCE, we have constructed a set of classes that addresses all three user-type groups (verbatim from the paper).
6. **Models Evolve.** The model needed in concept formulation is very different than the model needed in detailed design, or in operations. Models need to evolve and grow, and sometimes shrink. This should be the focus of model reuse along the project lifecycle. … It is clear that the more a model can be a self-contained, internally self-consistent, and an intuitive description of the concept, the more informative it will be. Moreover, the more the analysis can be separated from the model, the more reusable it will be (verbatim from the paper).
7. **First Description, Then Analysis.** The more the analysis can be separated from the model, the more reusable it will be. For our mass analysis we have achieved a high degree of separation of the model from the analysis, and as a result we are able to run exactly the same mass analysis script on all three of our mission option models (verbatim from the paper).
8. **Real Examples are Powerful.** Trying to describe to stakeholders and potential collaborators what MBSE looks and feels like has proven to be rather difficult and not very effective. We have found that many people 'get it' for the first time only when they see an actual example (verbatim from the paper).

My takeaway: leadership commitment is crucial; a core support team is crucial, one that both assists projects and trains project personnel; and model reuse is facilitated by modeling methods that separate analysis from models.

A later presentation on IMCE infusion at JPL (Lin 2014) addresses enablers, barriers, and challenges experienced at JPL, and provides status of IMCE.

Selected portions of relevance follow:

Infusion Enablers and Barriers

Enablers
- Innovative engineers
- Long development cycles
- Recent experience with challenges of project knowledge management
- Organizational combination of systems engineering and software engineering
- Strong management support

Barriers

- Conservative engineering philosophy due to unforgiving nature of the environments in which we operate
- Maturity of the integrated tools
- Big learning curve for experienced practitioners

Addressing Challenges

- Reticence to adopt new methods
  - Mitigation:
    - Use pilots to show value;
    - Work through the line organization to take ownership
    - Strategically embed MBSE-enabled engineers in projects as infusion agents
    - Vendor tool maturation to meet our needs
  - Mitigation:
    - Work with industry standards bodies and vendors to influence them to address our needs
    - Invest in developing tools that augment commercial solutions

Current State of MBSE: Institutional Capability

- Phase I (done)
  - Exposed a cadre of engineers (200+) to concept of system modeling; more than 50+ are in practice
  - Established a system modeling environment that consists of a commercial modeling tool and JPL SysML modeling guide
  - Completed, reviewed and deployed a set of system engineering modeling standards and a lexicon to be used for developing and structuring system models (e/g. ontologies, ontology-to-profile, profile-to-plugin)
  - Provided consultation services for 20+ projects applying IMCE developed capabilities
  - Automated document generation capability from the model
- Phase II (in progress)
  - Began to define a mechanism for data exchange between models (e.g., systems and subsystem)
  - Model repository (Teamserver) deployed and in use by multiple tasks and projects
  - Begin to capture re-usable modeling design patterns
- Phase III (piloting)
  - Explored and gained experiences in trade-space analysis with large variant space

My takeaway: enterprise transformation to a model-based engineering operational environment is necessarily a phased approach that incrementally learns from experience and experimentation; engineers will learn, value, and adopt system modeling approaches (if the understand a realistic beneficial value proposition); creation and utilization of local modeling standards is crucial; a core and competent support team for project consulting is necessary.

I.4.2. Boeing

Ron Carson, retired Boeing, presented a 2014 (and subsequently 2018) recounting of the MBSE infusion process at Boeing *(*Sheeley, Malone, Palmer, Carson 2014 and 2018). The summary discussed MBSE Process Needs, and MBSE Tool Needs, as follows:

Process Needs:

- Success stories – to help promote the benefits
- Methods to measure impact of MBSE
- Training to develop good "modelers"
- Methods to ensure persistence of MBSE after the advocates move on

Tool Needs:

- Lower user 'entry barriers' – more intuitive user interfaces
- Support for hundreds of globally distributed users
- Size scaling – consistent performance when managing large quantities of data and users
- Each diagram object is a database object
- Exchange and synchronization of federated engineering data (different databases)
- Other Engineering disciplines (SWE, EE, etc.)
- Transition to Manufacturing and PLM
- Tool vendors working together!
- Support for data reference libraries and data reuse
- Configuration and version control of all objects
- Bulk import / export / update capability

My takeaway: a support group that promotes benefits, reinforces values with evidence of usefulness, and provides training; and tools with low entry barriers, data federation, reusable data/model libraries, linkage to PLM, and scalability for large user and data bases.

I.4.3. Thales Group

From the abstract (Navas, Tannery, Bonnet, Voirin 2018): "This article presents a case study on applying model-based systems engineering (MBSE) methodologies under real-life conditions. We present how engineers tailored existing MBSE methods and tools to both address the complexity factors of nuclear power plants engineering, and to contribute to the comprehensiveness of the design and safety assessment. We also provide feedback on the application of MBSE approaches and their key benefits on projects' execution.

From the summary of findings and conclusion, verbatim:

> **FINDINGS**
>
> The MBSE-supported approach presented here is flexible enough to be applied to other kinds [of] NPP [Nuclear Power Plant] systems (fluid systems, simulation systems, and control and supervision systems, among others) at different stages of their design. The witnessed benefits include:

- A better communication and definition of responsibilities scopes between stakeholders: technical exchanges with detailed design teams, transversal disciplines such as safety or human factors, and other systems' architecture teams, were more productive when supported by common and normalized graphical representations provided by Arcadia/Capella.
- A unique source of information on systems architecture: Models encapsulate the key information and become the reference database for architecture-related topics, easing information capitalization for example, the extract of Interface Control Document tables.
- A fast learning curve: the Arcadia/Capella concepts and diagrams are rather well adapted to the nuclear engineering population mostly composed of engineers who [have] not been exposed to modelling approaches such as UML or SysML. The strong coupling between the method and the tool, and the availability of multiple productivity tools, are of great value for engineers.

> **CONCLUSION**
>
> This article showed how the application of MBSE methods and tools may have a positive impact on nuclear power plants' engineering task, and particularly during architectural definition and design. We witnessed benefits both in supporting the technical production, (by contributing to the exhaustiveness of design, safety justifications, and third-party assessments) and in the daily interactions between engineering teams (by providing a common and normalized graphical representation, and by introducing concepts that make teams work in a more collaborative and agile way). To guarantee these benefits, future users shall conjointly perform a tailoring of MBSE concepts and models to cope with their discipline and project-specific constraints.

My takeaway: Mixed-discipline engineering teams embraced values from graphical architecture models, inter-team communications, and mixed daily collaboration meetings; mixed-discipline engagement was facilitated by tools with fast learning curves.

I.4.4. Multiple Federal Agencies

An OSD-SE report on Digital Model-based Engineering: Expectations, Prerequisites, and Challenges of Infusion (MBSE Infusion Task Team 2017) offers challenges and recommended mitigation from an inter-agency task team (DoD, DHS, VA, FAA, and NASA).

> The team identified challenges an organization might encounter when looking to infuse DMbE:

- Assessing value added to the organization. Not all DMbE practices will be applicable to every situation in every organization; and not all implementations will have positive results.
- Overcoming organizational and cultural hurdles.
- Adopting contractual practices and technical data management.
- Redefining configuration management. The DMbE environment changes the range of configuration information to be managed to include performance and design

models, database objects, as well as more traditional book-form objects and formats.
- Developing IT infrastructure. Approaches to implementing critical, enabling IT infrastructure capabilities must be flexible, reconfigurable, and updatable.
- Ensuring security of the single source of truth.
- Potential over-reliance on quantitative data over qualitative data. Executable/computational models and simulations generally incorporate and generate quantitative vice qualitative data.

Prerequisites for the infusion of DMbE include management support/advocacy, technical capability readiness, and organizational/cultural willingness (or lack of resistance) to adopt a new methodology. Some level of management support is essential, and having a management champion or advocate is better still. This support may be gained through education and exposure to examples and benefits of DMbE. Encouraging and facilitating organizational and cultural change is often a challenge. Education, training, and access to the necessary tools, applications, and aids can be helpful. In general, lowering barriers to adoption and implementation is necessary. Helpful in all these cases is a clear statement or vision of a future state of the use of DMbE and of the approach or roadmap aligned with the vision going forward to identify avenues of infusion into normal business activities.

Organizations interested in infusing DMbE must recognize and identify the need and must be willing to make the necessary changes in established processes, tools and methods, and workforce. This results in a multifaceted approach that begins with the recognition that a change will have a positive outcome in the resultant capability, the staff makeup, and/or the speed of execution, with the expectation of higher precision and discovery of defects early on in a project's lifecycle. In other words, transition is a process, not an event.

It is understandable that the multifaceted approach will have to be planned and will not occur instantaneously. A willingness to change is accompanied by planning for the transition, identification of the stakeholder population among the adopters, and the understanding of what a successful change looks like.

My takeaway: necessary prerequisites to infusion include a clear vision of the future state, management support/advocacy, education and training, and lowering barriers to adoption and implementation is necessary. Importantly, a compelling need that can be appreciated by users must be identified.

I.5. Literature Reviews of Operational Requirements

I.5.1. SERC and Related MIT Research

A workshop was held in 2015 (Rhodes, Ross 2015) at the Massachusetts Institute of Technology in support of the SERC IMCSE program to investigate four interests (1) imagine an ideal world; (2) current state practice; (3) need for research; (4) emerging research; and (5) recommendations for gathering knowledge.

One participant offered this summary level statement as to the ideally envisioned experience of the individual: “An intuitive experience that generates deep insights across the area of relevant decisions that balances time, resources and the desired confidence in the decision outcome.”

Key themes of relevance to me emerged in the topic “Imagine an Ideal World,” with verbatim selections from the report as follows:

> **Ease of Interaction.** The individual interacting with a model will find it intuitive and the effort involved will be commensurate with the value the model provides. Novice users will be able to rapidly learn and benefit from use of modeling environments.
>
> **Enabling Human-Human Interaction.** Model-centric environments will support collaborative decision making and design with near real-time human to human interaction.
>
> **Guided Interaction.** Interactions with models will provide guided assistance for viewing models from standpoint of other stakeholders. … There will be assisted capabilities and wizards for model library curation, model composability, model interrogation, and stakeholder role playing.
>
> **Model Re-Usability.** The environment will be adaptable for the culture of the organization to enable effective reuse with confidence in the model and its appropriateness for the situation. Finding suitable models and reusing them in the individual’s unique model-based environment will be easily accomplished. … Effective digital curation will enable preserving, discovering and reusing appropriate models.

My takeaway: aptly summarized in the four bullet points above.

Many additional papers and presentations related to the SERC IMCSE project by Donna Rhodes, most with coauthors, were reviewed for relevance (see bibliography). Generally they go into research detail of the four key themes above, without additional points to make in this summary.

A non-SERC paper by Donna Rhodes (Rhodes 2018) does offer some new points of relevance. From that paper, verbatim:

> A common approach to undertaking a model-centric program is to send all the engineers through tool training. Younger engineers may quickly learn to use modeling tools, but lack the experiential knowledge of engineering of products and systems. The members with years of experience, on the other hand, may find use of modeling software tools to be non-intuitive to the point where they spend more time on tool mechanics and less on decision making. Any discomfort and distrust of models and modeling toolsets can have negative impacts on the engineering effort.

My takeaway: reinforcement of thought expressed frequently in the other reviews that tool ease of learning and use is especial crucial for older engineers with cultural legacy.

I.6. Tool Vendor Interview with PTC

Research than moved on to investigating the current plans of tool suppliers for integrated modeling platforms. The intention was to interview knowledgeable people at both PTC and Dassault (acquired NoMagic/MagicDraw) to ascertain the state of interoperable or integrated modeling and simulation tools other than SysML. Mathew Hause, of PTC agreed to spend three days overnight with me in February, 2018 on this discussion in a rental property on South Padre Island, TX that my family was occupying and was a reasonable drive from his home in Austin, TX. Matthew Hause has a recognized and deserved non-biased, non-selling, reputation for educating the community at large on MBSE and related standards (he sits on a number of MBSE-related standards committees). At the completion of the interview I did not feel a need to find someone as educationally un-biased at Dassault. Hause was quick to point out that a lot of what PTC is doing for tool-interoperability is also being done with Dassault's Cameo as well.

My tool vendor interview was motivated by a strong supposition that ARDEC, based on verbal comments made at an ARDEC/RT168 review meeting, would be seeking a commercially available MBSE platform with improved integration of multi-tool capabilities over what is currently available. One objective was to understand how RT168 efforts, and my task in particular, might be usefully adopted by tool vendors.

Hause made it very clear that tool vendors are not interested in integrated-tool platform environments, but rather have high interest in tool and data interoperability.

Standards work on Open Services for Life Cycle Collaboration (OSLC) provides interoperability linkage and methods used by many tool vendors at this point. Attempts to find a good public body of knowledge on OSLC is not highly rewarding – the results and ongoing efforts appear to be principally available to the companies involved in the standards work. However, an excellent overview is available at (Szarazi 2014).

My principle takeaway: Multi-tool interoperability is further along than I had suspected, principally due to standards work on Open Services for Life Cycle Collaboration (OSLC).

I.7. Discussion and Opinion

Basic early versions of Model Centric Engineering Platforms (MCEPs) are available from commercial suppliers such as PTC and Dassault (with the acquisition of No Magic). More advanced MCEPs are the work-in-process of various research and feature-prototyping projects, such as RT-168; partly to overcome vendor-centric issues and limitations, but mainly to innovate new capabilities.

Research and prototyping work is generally focused on functional features: interoperability infrastructure, interoperability with new tools and applications, single-source-of-truth, knowledge of conflicts among engineering activities in process, trade-off and decision making support, and stakeholder-relevant communications and progress status.

To date, little has been focused on the operational requirements. Operational requirements include MCEP facilitation of configuration engineering, sustainment, evolution, and stakeholder/user engagement. This task focused on research into user engagement requirements. Requirements identified in this summary are at the capability level rather than the feature level, with the intent to actionably inform a variety of MCEP interests including researchers, feature prototype developers, MBE platform developers and suppliers, and tool-acquisition decision makers.

Three bodies of knowledge have principally guided the research here: agile systems engineering operations, Live-Virtual-Constructive (LVC) operations, and software IDE platform operations.

I.7.1. Software IDE relevance.

There are many differences between the nature of a systems engineering platform (MCEP) and a software engineering platform (IDE), but both necessarily share many common capabilities for configuration, operation, and life cycle support.

I.7.2. LVC relevance.

Live-Virtual-Constructive is an architecture-enabled operational concept more than a platform. LVC concepts are associated by many with DoD training and testing environments that mix simulations, people, and operable equipment. But the general concept of mixed L-V-C environments has employment and history in other areas, most notably for System Integration Laboratories that start with simulations and people and incrementally replace simulations (and some people) with proxy or finished equipment, e.g., (Dove, Schindel, Garlington 2018).

L-V-C is explained below from Wikipedia excerpts

> **Live** - A simulation involving real people operating real systems. Military training events using real equipment are live simulations. They are considered simulations because they are not conducted against a live enemy. (Wikipedia)
>
> For MCEP, real systems and real people involved in a systems-of-systems interoperable development and/or test environment (hence, a simulation). These real elements may be operational prototypes, available lower fidelity versions of systems under development, and people that interact with these systems in any way under operational conditions.
>
> **Virtual** - A simulation involving real people operating simulated systems. Virtual simulations inject a Human-in-the-Loop into a central role by exercising motor control skills (e.g., flying jet or tank simulator), decision making skills (e.g., committing fire control resources to action), or communication skills (e.g., as members of a C4I team). (Wikipedia)
>
> **Constructive** - A simulation involving simulated people operating simulated systems. Real people stimulate (make inputs to) such simulations, but are not involved in determining the outcomes. A constructive simulation is a computer program. For example, a military user may input data instructing a unit to move and to engage an enemy target. The constructive simulation determines the speed of movement, the effect of the engagement with the enemy and any battle damage that may occur. These terms should not be confused with specific constructive models such as Computer Generated Forces (CGF), a generic term used to refer to computer representations of forces in simulations that attempts to model human behavior. CGF is just one example model being used in a constructive environment. There are many types of constructive models that involve simulated people operating simulated systems. (Wikipedia)

I.7.3. Agile Systems Engineering relevance.

The bulk of this research report employs agile system and system engineering fundamental understandings. Partly because it has been the author’s life work since 1991, but mainly because it has direct and comprehensive application relevance to the operational aspects of an MCEP.

I.7.4. Value Proposition

An MCEP facilitates the development and evolution throughout the life cycle of an SoS digital twin. It provides an integrated SoS model for systems engineers. It provides an integration test and evaluation environment for component engineers. It provides a training environment for operations personnel. It provides a source of current knowledge and state-of-roduct for maintenance personnel. It lives throughout the SoS life cycle. At core, it supports planning, design, and tradeoff decisions with a single source of truth. It's principle values are in the reduction of rework in all life cycle stages and in clear stakeholder communications.

I believe the core values of iMBE/MCEP are in rework reduction, stakeholder communications, life cycle support from a digital twin, and both SoS (product) and operational (process) agility. Among other things, those core values elaborate to decision and trade-off support, and single source of truth throughout the life cycle. Rework reduction appeals to the customer, the SE, and the multi-discipline engineers (where ARDEC is getting pushback – likely because the value proposition pitch is heard as something completely different instead of a naturally welcome productivity enhancement easily assimilated as an augmentation to current practice).

I.8. References

Bayer, T. J., M. Bennett, C. L. Delp, D. Dvorak, J. S. Jenkins, S. Mandutianu. 2011. Update – Concept of Operations for Integrated Model-Centric Engineering at JPL. 2011 Aerospace Conference, IEEE. Big Sky, MT, USA, 5-12 March. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.667.4638&rep=rep1&type=pdf

Bayer, T, S, Chung, B. Cole, B. Cooke, F. Dekens, C. Delp, I. Gontijo, K. Lewis, M. Moshir, R. Rasmussen, D. Wagner. 2012. Early Formulation Model-Centric Engineering on NASA's Europa Mission Concept Study. INCOSE International Symposium, Rome, Italy, 9-12 July. https://trs.jpl.nasa.gov/bitstream/handle/2014/42873/12-1191_A1b.pdf

Dove, R., W. Schindel, and K. Garlington. 2018. "Case Study: Agile Systems Engineering at Lockheed Martin Aeronautics Integrated Fighter Group." Paper presented at the International Council on Systems Engineering, International Symposium, Washington, US-DC, 7-12 July. www.parshift.com/s/ASELCM-04LMC.pdf.

Goebel, R,, C. Pavlakos. 2012. A Common Engineering Environment (CEE) Implementation. Sandia National Laboratories, Report SAND2012-4154 C. https://osti.gov/scitech/servlets/purl/1117233&usg=AOvVaw0aozgFaWsHvNf1T0tVxsDO

Lin, C. Y. 2014. Presentation–Integrated Model-Centric Engineering: Infusing MBSE at JPL. INCOSE International Workshop, MBSE Workshop. Torrance, CA, January 25-26. www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:incose_mbse_iw_2014:mbse_infusion-jpl.pptx

MBSE Infusion Task Team. 2017. Digital Model-based Engineering: Expectations, Prerequisites, and Challenges of Infusion. Interagency Working Group on Engineering Complex Systems (IAWG), Office of the Deputy Assistant Secretary of Defense, Systems Engineering. March. https://www.acq.osd.mil/se/docs/DMbE-20170524-0602.pdf

Navas, J., P. Tannery, S. Bonnet, J. Voirin. 2018. Bridging the Gap Between Model-Based Systems Engineering Methodologies and Their Effective Practice – A Case Study on Nuclear Power Plants Systems Engineering. INCOSE INSIGHT 21(1):17-20, March. https://onlinelibrary.wiley.com/doi/epdf/10.1002/inst.12185, or with INCOSE membership: https://connect.incose.org/Library/InsightMagazine/

Pavlakos, C. 2014. Common Engineering Environment (CEE), Sandia National Labs. November. Presentation made to and available from Mark Blackburn.

Rhodes. D. H., Ross, A. M. 2015. Interactive Model-Centric Systems Engineering (IMCSE) Pathfinder Workshop Report. Systems Engineering Research Center. 28 February. www.dtic.mil/get-tr-doc/pdf?AD=ADA626620

Rhodes, D. H. 2018. Using Human-Model Interaction Heuristics to Enable Model-Centric Enterprise Transformation. IEEE International Systems Conference (SysCon). Vancouver, BC, Canada, April 23-26

Schindel, W., R. Dove. 2016. Introduction to the Agile Systems Engineering Life Cycle MBSE Pattern. Proceedings International Symposium. International Council on Systems Engineering. Edinburgh, Scotland, 18-21 July. www.parshift.com/s/160718IS16-IntroToTheAgileSystemsEngineeringLifeCycleMBSEPattern.pdf

Sheeley, B., R. Malone, J. Palmer, R. Carson. 2014. MBSE Implementation Across Diverse Domains at The Boeing Company. Unpublished presentation made to INCOSE MBSE Working Group, January; subsequently made to INCOSE Enchantment Chapter 14 February 2018, www.incose.org/docs/default-source/enchantment/180214carsonron-mbseimplementationacrossdiversedomainsattheboeingcompany-slides.pdf?sfvrsn=2

I.9. Bibliography (selected influential literature sources reviewed but not cited in this summary)

Bajaj, M., D. Zwemer, R. Yntema, A. Phung, A. Kumar, A. Dwivedi, M. Waikar. 2016. MBSE++ — Foundations for Extended Model-Based Systems Engineering Across System Lifecycle. Proceedings IS16 Symposium. International Council on Systems Engineering. Edinburgh, Scotland, UK, July 18-21.

Clement, S., D. McKee, R. Romano, J-M. Lopez, D. Battersby. 2018. The Internet of Simulation: Enabling Agile Model Based Systems Engineering for Cyber-Physical Systems. System of Systems Engineering Collaborators Information Exchange Webinar. DASD(SE) and NDIA SE Division. March 6.

DoD. 2018. Digital Engineering Strategy. Office of the Deputy Assistant Secretary of Defense for Systems Engineering.

Hale, J. P. Gill. 2017. Presentation: Moving NASA/MSFC towards a more Model-centric Organization. NASA/MSFC Office of Chief Engineer, Systems Engineering Management Office. May 2-3, 2017

Kennedy, M. 2012. Set-Based Decision Making – Taming System Complexity. Lean Software and Systems Conference. Boston, MA May 13-18. Video at https://vimeo.com/42785298

Kennedy, B. M. D. K, Sobek, M. N. Kennedy. 2014. Reducing Rework by Applying Set-Based Practices Early in the Systems Engineering Process. Systems Engineering 17(3). Wiley.

LiCausi, T. 2018. Modeling, Simulation, and Analysis Fundamentals. INCOSE Webinar, January 17. (Live/Virtual/Constructive)

Reid, J. B., D. H. Rhodes. 2016. Digital System Models: An investigation of the non-technical challenges and research needs. Conference on Systems Engineering Research. Huntsville AL. March 22-24.

Rhodes, D. H., Ross, A. M. 2016. Interactive Model-Centric Systems Engineering (IMCSE) Phase 3, Technical Report SERC-2015-TR-043-1. Stevens Institute of Technology, Systems Engineering Research Center. March 1.

Rhodes, D. H. 2017. Why is Human-Model Interactivity Important to the Future of Model-Centric Engineering? NASA/JPL Symposium & Workshop on Model-Based Systems Engineering. January 26.

Rhodes, D. H., Ross, A. M. 2017. Interactive Model-Centric Systems Engineering (IMCSE) Phase 4, Technical Report SERC-2017-TR-103. Stevens Institute of Technology, Systems Engineering Research Center. March 1.

Rhodes, D. H. 2018. Interactive Model-Centric Systems Engineering (IMCSE) Phase 5, Technical Report SERC-2018-TR-104. Stevens Institute of Technology, Systems Engineering Research Center. February 28.

Rhodes, D. H. 2018. Using human-model interaction heuristics to enable model-centric enterprise transformation. Annual IEEE International Systems Conference (SysCon). Vancouver, BC, Canada, April 23-26.

Szarazi, J. 2014. Presentation: Continuous collaboration in engineering through OSLC. October 7. www.crystal-artemis.eu/fileadmin/user_upload/Other/Jerome_Szarazi_Koneksys_Continuous_collaboration_in_engineering_through_OSLC_Oct2014.pdf

Theiss, M. 2017. 3DExperience Interoperability. Prostep AG. http://webinarsbyprostep.com/wp-content/uploads/2017/11/Brian-Schouten-3DX-Integration-and-Migration-COE-2017-Workshop.pdf

Zimmerman. P. 2014. Presentation: Digital System Model Development and Technical Data. 17th Annual NDIA Systems Engineering Conference, Springfield, VA, October 30.

Zimmerman. P. 2015. Use of Data Models for Department of Defense Acquisition. Unified Architecture Framework & Model Based Systems Engineering Information Day. Reston, VA, March 23

Zimmerman. P. 2015. A Framework for Developing a Digital System Model Taxonomy. 18th Annual NDIA Systems Engineering Conference, Springfield, VA, October 28.

J. LIST OF MEETINGS, DEMONSTRATIONS, DELIVERABLES AND ARDEC-RELEVANT EVENTS

Table 1 provides a list of the deliveries, demonstrations and discussions for our bi-weekly status and other meetings involving our ARDEC sponsors. Again, for ease of assessment and continuity, we have prepended the Phase II demonstrations and deliverables to those of Phase I in Table 1.

Table 1. Schedule for Demonstration and Deliverables

Date	Phase II: Demo / Presentation /Reports	Status

Aug 21, 2017	ARDEC next year plans, including organizational changes, led by Eddie Bauer	Done
Aug 30, 2017	Graphical CONOPS “integrated” with Rhapsody-driven (System Model) Simulations: presented by Omar Valverde, MITRE	Done
Aug 30, 2017	"Architecture and Prototyping of System Simulation with Semantic Data Exchange, presented by Kishore Pochiraju	Done
Sep 3, 2017	Meeting notes for working session #5	Done
Sep 14, 2017	Eddie Bauer: AVCE iMBE kick-off, led by Eddie Bauer	Done
Sep 20, 2017	AVCE-iMBE SysML model and analysis, presented by Rick Dove	Done
Sep 27, 2017	Using MDAO workflows to formalize Assessment Flow Diagram (ASD) of Decision Framework concept by Matt Cilli, & instantiated in AAMODAT, presented by John Dzielski	Done
Oct 11, 2017	A009 – RT-168 Technical Management Plan	Done
Oct 24, 2017	Meeting notes for working session #5	Done
Oct 25, 2017	Briefing of ARDEC and NAVAIR research at National Defense Industry Association (NDIA) Systems Engineering Conference, presented by Mark Blackburn	Done
Nov 8, 2017.	Briefing of ARDEC and NAVAIR research at SERC Sponsor Review, presented by Mark Blackburn	Done
Nov 14, 2017	Update on Decision Framework and Formalizing Assessment Flow Diagram (ASD) through MDAO, presented by John Dzielski	Done
Nov 15, 2017	A008 RT-168 Bi-monthly status	Done
Nov 28, 2017	ANSYS on a workbench platform for integrating tools on that supported the ARDEC/ANSYS Developed Analysis Preprocessing Tool (ADAPT), presented by Ryan Gordon	Done
Dec 5, 2017	Ontology Bootcamp, presented by Barry Smith, video: https://www.youtube.com/watch?v=bj8mSbHh-qA&list=PLyngZgIl3WTgK3qMmOWt4VDIbh-xB3Ejkz	Done
Dec 6, 2017	AVCE iMBE IPT Meeting to discuss changes due to Eddie Bauer transitioning to another assignment	Done
Dec 18, 2018	Dr. Thomas Hagedorn successfully defended his final dissertation defense. His advisor is Dr. Ian Grosse from the Univ. of Massachusetts	Done
Jan 12, 2018	A008 RT-168 Bi-monthly status	Done
Feb 7, 2018	Applications for Three Research Use Cases in Model Centric Engineering Using ModelCenter and MBSEPak, video available at Phoenix Integration	Done

Feb 21, 2018	Demonstration of OpenMBEE, which is installable through Docker script and now running on Amazon Web Services and Stevens servers	Done: updates continuous
Ongoing	Source code for Integration and Interoperability Framework (IoIF) has been provide to the ARDEC sponsors	Done
Mar 15, 2018	A013 – RT-168 Interim Technical Report for Phase II	Done: this report
Apr 3, 2018	Working Session #9	Done
Apr 10, 2018	SERC Advisory Board at MITRE in McLean, VA	Done
Apr 18, 2018	Present research at Phoenix Integration Users Conference on Three Application of MDAO, by Mark Blackburn	Done
May 15, 2018	A008 – RT-168 Bi-monthly status report to SERC (Mar/Apl)	Done
Jun 5, 2018	Working Session #10	Done
Jul 15, 2018	A008 – RT-168 Bi-monthly status report to SERC (May/Jun)	Done
Jul 31, 2018	Working Session #11	Done
Aug 8, 2018	Uploaded using AMRDEC Safe tools, models, and software	Done
Aug 8, 2018	A013 – RT-168 Final Technical Report for Phase II	Done
Date	**Phase I: Demo / Presentation / Reports**	**Status**
Sep 21 & 22, 2016	1st Working Session at ARDEC – see meeting notes.	Done
Nov 4, 2016 (Fri)	Mission Level Modeling and Graphical CONOPS (2 approaches) - *Paul Grogan* - *Roger Blake* *Roger Jones*	Done
Nov 7, 2016	Interim Report/Bi-Monthly Status - Expand on all tasks that are mapped to Use Cases project model	Done
Nov 22, 2016	Decision Framework Approach by *Matt Cilli / Robin Dillon*	Done
Dec 2, 2016	MDOA presentation and demonstration by *Steven Hoffenson* *Discussion of Mission/System Simulations Roger Jones, Roger Blake, Paul Grogan*	Done
Dec 16, 2016	Design of a Systems Representation Framework for Counter UAS Operations *Kishore Pochiraju*	Done
Dec 20, 2016	Information Model/Ontology by *Mark Blackburn / Mary Bone / Gregg Vesonder*	Done
Jan 10, 2017	2nd Working Session at ARDEC – see meeting notes.	Done
Jan 15, 2017	Update Interim Report/Bi-Monthly Status - Expand on tasks that are mapped to use cases in project model	Done: This report
Jan 25-27, 2017	NASA/JPL Symposium and Workshop on Model Based Systems Engineering	Meeting notes delivered

Jan 28-31, 2017	INCOSE International Workshop	Meeting notes delivered
Feb 10, 2017	Demonstrations of Graphical CONOPS ▪ Roger Jones – Unity gaming of competing autonomous quadcopters ▪ Todd Richmond – Video of Unity gaming for Early Synthetic Prototyping	Done
Feb 24, 2017	Automatic Concurrent Engineering and Knowledge-Based Product Design and Manufacturing (Kishore Pochiraju)	Done
Mar 2, 2017	Semantic Web Technologies (Mary Bone / Mark Blackburn)	Done
Mar 7, 2017	Syndeia Demonstration (Manas Majaj / Jeff McDonald)	Done
Mar 9, 2017	ARDEC sponsor Eddie Bauer participated in NAVAIR, RT-170 working session #29 at NAVAIR.	Done
Mar 10, 2017	Update on HLA approach (Roger Blake / Paul Grogan)	Done
Mar 15, 2017	Update Interim Report Expand on tasks that are mapped to use cases in project model	Done: Prior version of this report
Mar 24, 2017	Mary Bone gave a talk on ontologies as it related to AAMODAT and Challenge area #5	Done
Mar 30, 2017	Working Session #3 at Stevens – see meeting notes. There were over 25 attendees, including nine (9) from ARDEC	Done
Apr 7, 2017	Kishore gave a talk on Design Automation	Done
Apr 18, 2017	Two related talks on OpenMBEE model in SysML to support analysis of requirements development/review for AVCE iMBE (Mark Blackburn)	Done
Apr 21, 2017	Broader aspects of OpenMBEE (Mark Blackburn)	Done
May 15, 2017	Bi-monthly status report • Expand on tasks that are mapped to use cases in project model	Done
May 19, 2017	Model Centric Engineering Architecture (Roger Blake/ Paul Grogan)	Done
Jun 2, 2017	Overview on Model Development Kit (MDK) DocGen View and Viewpoints that were added to AVCE requirements model to illustrate the DocGen capabilities (Benjamin Kruse)	Done
Jun 13, 2017	Working Session #4 at ARDEC – see meeting notes.	Done
Jun 30, 2017	Two talks on Model Centric Engineering Architecture and the Prototype of the Integration and Interoperability Framework (IoIF) and demonstration interoperability using	Done

	semantic web technologies and ontologies (Paul Grogan, Roger Blake, Mary Bone, Chris Synder, Harsh Kevadia)	
Jul 14, 2017	Decision Framework update with discussion of use of semantic web technologies and concept for modeling the Assessment Flow Diagram (Matt Cilli, Robin Dillon-Merrill, Mary Bone, John Dzielski)	Done
Jul 15, 2017	Updated Interim Report • Expand on tasks that are mapped to use cases in project model	Done
July 31, 2017	Systems Engineering Transformation through Model Centric Engineering Past, Present, and Future – Special Session at Stevens (Mark Blackburn, Dinesh Verma)	Done
Aug 1, 2017	Working Session #5 at Stevens	Done
Aug 8, 2017	Final Technical Report	Done