NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CONCEPT OF OPERATIONS
FOR THE ENTERPRISE ENGINE

by

Scott J. Rosa

September 2018

Thesis Advisor: Dan C. Boger
Co-Advisor: Scot A. Miller

Approved for public release. Distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** September 2018	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE CONCEPT OF OPERATIONS FOR THE ENTERPRISE ENGINE	**5. FUNDING NUMBERS**
6. AUTHOR(S) Scott J. Rosa	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

Engineering design and development processes have evolved from hand-drawn sketches to complex, digital models across all fields of engineering. After researching this evolution, the Department of Defense (DoD) should embrace this transition and further invest in model-based development in its pursuit of future complex systems of systems (SoS). The Enterprise Engine (EE) offers a development, integration, and process improvement approach that appears faster, cheaper, and smarter while enabling semantic interoperability that has the potential to cause a revolution in DoD acquisition. EE is the vision of several masters of engineering development; however, all their ideas have not been captured in a cogent, easy-to-understand concept of operations. This thesis provides the background, concepts, and analysis of incorporating EE in future software development and presents an executable CONOPS to integrate EE within the current acquisitions environment and practices.

14. SUBJECT TERMS MBSE, model-driven development, interoperability	**15. NUMBER OF PAGES** 89
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

CONCEPT OF OPERATIONS FOR THE ENTERPRISE ENGINE

Scott J. Rosa
Major, United States Marine Corps
BS, U.S. Naval Academy, 2007

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2018

Approved by: Dan C. Boger
Advisor

Scot A. Miller
Co-Advisor

Dan C. Boger
Chair, Department of Information Sciences

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Engineering design and development processes have evolved from hand-drawn sketches to complex, digital models across all fields of engineering. After researching this evolution, the Department of Defense (DoD) should embrace this transition and further invest in model-based development in its pursuit of future complex systems of systems (SoS). The Enterprise Engine (EE) offers a development, integration, and process improvement approach that appears faster, cheaper, and smarter while enabling semantic interoperability that has the potential to cause a revolution in DoD acquisition. EE is the vision of several masters of engineering development; however, all their ideas have not been captured in a cogent, easy-to-understand concept of operations. This thesis provides the background, concepts, and analysis of incorporating EE in future software development and presents an executable CONOPS to integrate EE within the current acquisitions environment and practices.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AoA	analysis of alternatives
AML	archetype modeling language
API	application programing interfaces
CBA	capabilities-based assessment
CDD	capability development document
CDR	critical design review
CIM	computation independent model
CONOPS	concept of operations
CPD	capability production document
DAS	Defense Acquisition System
DEVOPS	development operations
DNA	deoxyribonucleic acid
DoD	Department of Defense
DoDAF	Department of Defense Architecture Framework
DOTMLPF	Doctrine, Organization, Training, Material, Leadership, Personnel, and Facilities
EE	Enterprise Engine
EMD	engineering & manufacturing development
ICD	initial capabilities document
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council on Systems Engineering
IOT&E	initial operational test and evaluation
JCIDS	Joint Capabilities Integration and Development System
JPL	Jet Propulsion Laboratory
KPP	key performance parameters
LCIM	Levels of Conceptual Interoperability Model
LCSP	life cycle sustainment plan
LIDAR	light detection and ranging
LRIP	low-rate initial production
M&S	modeling and simulation

MARCORSYSCOM	Marine Corps Systems Command
MBE	model-based engineering
MBSE	model-based systems engineering
MCWL	Marine Corps Warfighting Laboratory
MDA	Model Driven Architecture
MDD	model-driven development
MDK	Model Driven Development Toolkit
MOF	Meta Object Facility
MOSA	modular open systems approaches
MSA	material solution analysis
MSL	Mars Science Laboratory
NAVAIR	Naval Air Systems Command
NAVFAC	Naval Facilities Engineering Command
NAVSEA	Naval Sea Systems Command
NAVSUP	Naval Supply Systems Command
NDIA	National Defense Industrial Association
NPS	Naval Postgraduate School
NtM	Notice to Mariners
O&S	operations and support
OMG	Object Management Group
ONR	Office of Naval Research
P&D	production and deployment
P&R	programs and resources
PCF	Pivotal Cloud Foundry
PDR	preliminary design review
PIM	platform independent model
POTUS	President of the United States
PPB&E	Planning, Programming, Budgeting, and Execution
PSM	platform specific model
QDR	Quadrennial Defense Review
RCPI	Rapid Capabilities Provisioning and Integration
RMF	Risk Management Framework

SEP	systems engineering plan
SME	subject-matter expert
SoS	systems of systems
SPAWAR	Space and Naval Warfare Systems Command
SPO	systems program office
SysML	system modeling language
TEMP	test and evaluation master plan
TMRR	technology maturation and risk reduction
TRR	test readiness review
TSE	trade space exploration
TTP	tactics, techniques and procedures
UAV	unmanned aerial vehicle
UGV	unmanned ground vehicle
UML	Unified Modeling Language
UTACC	Unmanned Tactical Autonomous Control and Collaboration

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would first like to thank my wife, Alison, for her never-ending support and patience during the many hours she spent holding things together for our family. I would also like to thank my children, Charles, Lauren, and Jonathan, for always bringing a smile to my face and keeping me motivated to complete this thesis. Next, I would like to thank my parents, Stephan and Glenda, and my sisters, September and Stephanie, for always rooting me on and giving me inspiration.

I would also like to thank my thesis advisor and co-advisor, Dr. Dan Boger and Scot Miller, respectively, for their guidance, patience, and efforts in assisting me throughout the thesis process. I could not have finished this thesis without your support and mentorship.

I would also like to thank several experts and subject-matter experts who greatly assisted with the technical knowledge within my thesis. Norm Eaglestone, Professor William Fast, and Manfred Koethe provided many hours of their time helping me understand the Enterprise Engine and the Defense Acquisitions System.

Finally, I would like to thank Alison Scharmota and Janice Long for amazing support by helping with editing and formatting my thesis.

This thesis could not have been possible without everyone's help.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The Department of Defense's (DoD) emphasis on innovation and capability enhancements continues increasing due to the proliferation of data-, information- and knowledge-based technologies and the increased importance of integrated systems of systems (SoS). The DoD struggles to meet these emerging requirements in a timely and cost efficient manner for several reasons.

A. CHALLENGES

First, the outdated documents-based development process results in minimal reuse of previously developed software. This inability causes information siloes, delayed product delivery, and increased development costs.

Second, system interoperability and integration gaps have emerged due to systems becoming more complex and the increase of reliance on SoS. Even when the DoD agrees on using one set of standards, such as the J-series messages used in Link-16, implementation of that standard varies by platform, still creating interoperability issues.

Third, there is a continuum of levels of interoperability. The DoD has, at best, accomplished syntactic interoperability, meaning that various systems understand the message formats that exchange between them. However, the DoD should strive for at least semantic interoperability, where the meaning of the information contained in those messages can be shared across systems. This is an imperative for an organization that trumpets developing products that drive "data to decisions."

Finally, the DoD faces increasing cyber security issues. In the past, systems implemented security capabilities as an afterthought, and many of these controls were rendered obsolete rapidly after initial operating capability. Today the DoD requires the development of cyber security as an integral part of the system and requires continuous monitoring and security upgrades. However, the DoD lacks efficient approaches to achieve those worthy goals.

These emerging problems have driven a Naval Postgraduate School (NPS) research team, comprised of information technology and software development researchers, to examine these challenges and develop possible solutions to address them.

The NPS research team's proposed solution is called the Enterprise Engine (EE). The EE is an enterprise-wide life-cycle process consisting of six main components. These components focus on maximizing the use of formal modeling via model-driven development (MDD), semantic enabling, automated code generation, code quality and cyber-health, a Rapid Capabilities Provisioning and Integration (RCPI) platform, and deployment. The EE, if developed and fully integrated into the software life cycle, addresses many current capability gaps in the present development environment. The researchers, aided by various development teams, confirmed the EE concept in six successful proofs of concept. Industry success examples also exist.

B. METHODOLOGY

The qualitative research method that developed and accomplished this project entailed comprehensive research of the important concepts and tools within the scope of employment of the EE. These areas included model-based systems engineering (MBSE), model-driven development (MDD), ontologies, information security, and the Defense Acquisition System (DAS), along with other interrelated topics as required throughout the research.

C. PROBLEM

1. Problem Statement

The research problem this study addresses is that there is no formal concept of operations (CONOPS) for the implementation of the EE in order to leverage its tremendous potential.

2. Purpose Statement

The purpose of this research is to develop and propose a CONOPS for the EE that guides potential EE stakeholders through its processes and makes recommendations on its implementation. Executing this research meant learning and applying acquisition processes, understanding EE tool capabilities, and mapping the EE into various stages of the acquisition process and timeline. Qualitative research methods were the primary means of research to examine and understand the key components of the EE, survey the current military acquisitions environment, and finally develop a feasible CONOPS for the employment of the EE.

D. CHAPTER OUTLINE

This thesis is organized into five chapters:

Chapter I introduces the EE, the problem and purpose of the research, and outlines the thesis objectives.

Chapter II provides a literature review related to the development, integration, and security approaches the EE provides, a summary of the acquisitions process and a basic review of the importance of CONOPS.

Chapter III describes the research methodologies of this thesis to achieve the purpose of proposing a formal CONOPS for the EE.

Chapter IV proposes a CONOPS for the implementation of the EE.

Chapter V provides a summary of the thesis and proposes recommendations for further research pertinent to this subject.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

This literature review describes the foundation and basic approaches the EE incorporates in efforts to improve development, integration, and security implementation within a SoS. These approach descriptions will provide necessary background research pertaining to the main concepts behind EE components and tools. Following this overview, essential background research pertaining to the DoD acquisitions process is provided along with a description of the applicability and necessity for a CONOPS.

A. EE DEVELOPMENT APPROACH

The EE's developmental approach relies heavily upon the building and development of formal models using the fundamentals of MBSE and MDD. During formal model development, parametric modeling and simulation incorporates analysis of alternatives, what-if analysis, and enables developmental tests. These tests are combined with standards-based development and modeling transforms which allows for models to operate under common, adaptable structures that maximize use of previous models within the SoS. This model-driven development approach fosters the compilation of ontologies, which results in greater levels of interoperability, integration, and provisioning. It also enables the maximization of automated code generators, developing up to 16,000 lines of secure code per minute, ultimately allowing for "humans to do what humans do best and machines to do what machines do best" (N. Eaglestone, interview with author, October 11, 2017).

1. MBSE Basics

The EE's foundation is built upon MBSE and model-based engineering (MBE) fundamentals. Both MBSE and MBE primarily rely upon the building and use of models throughout a system's complete development, implementation, and life cycle. The International Council on Systems Engineering ([INCOSE] 2007) defines MBSE as "the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout the development and later life cycle phases" (p. 15). Similarly, the

National Defense Industrial Association (NDIA) defines MBE as "an approach to engineering that uses models as an integral part of technical baselines that includes the requirements, analysis, design, implementation, and verification of a capability, system, and/or product throughout the acquisition life cycle" (NDIA Systems Engineering Division, M&S Committee, 2011, p. 9). For simplicity, this thesis will primarily use the term MBSE throughout.

Electrical and mechanical design have effectively used models-based approaches to engineering since the 1980s (Friedenthal, Moore, & Steiner, 2008, pp. 16–17). Figure 1 is a simple illustration of how mechanical engineering transitioned from a paper-based design process to a fully automated, computer- and model-based design process.

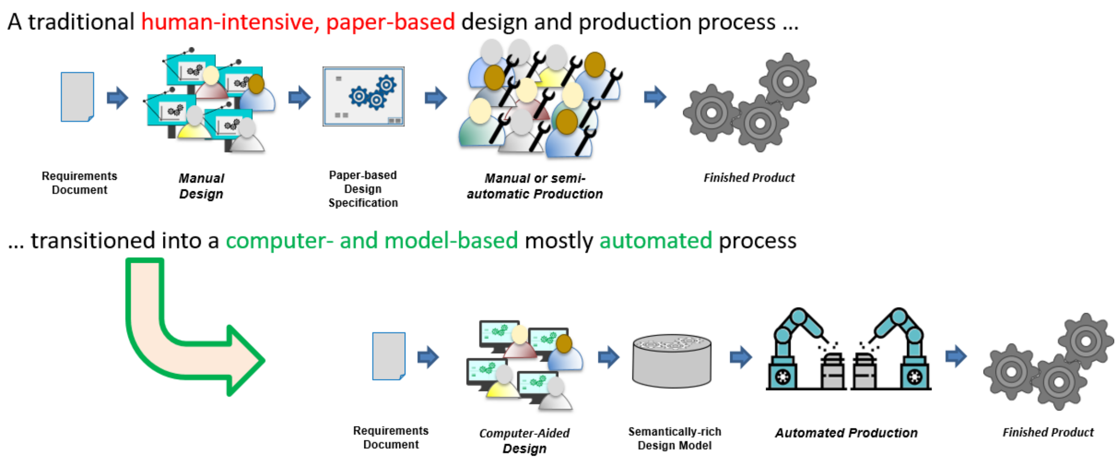


Figure 1. Computer- and Model-Based Process versus Human-Intensive, Paper-Based Process. Source: M. Koethe, personal communication (2017).

MBSE expands upon the early established foundations of systems engineering that primarily used documents-based methods to complete necessary engineering activities and incorporates the conceptual model-based ideals that the mechanical and electrical engineering fields utilize (Friedenthal et al., 2008, p. 17). Modernization of systems engineering practices toward this model-based concept allows for improved communications between designers and stakeholders, increased ability to oversee progressively complex systems, and standardized knowledge capture and rapid reuse (Walden et al., 2015, p. 189).

The implementation and use of MBSE practices have steadily gained traction throughout commercial industries and governmental agencies due to advantageous gains in the development effectiveness and efficiencies. While there are countless examples of MBSE's use, two high-profile use-cases exemplify the trend. The first of these cases outlines the implementation and use of MBSE for a complex space project and reinforces the advantages associated with its use. The second case demonstrates the large-scale implementation of MBSE across a large industry-leading company. These use-cases provide insight and lessons learned that benefit the CONOPS for EE.

The first case is the Jet Propulsion Laboratory's (JPL) and California Institute of Technology's Mars 2020 project. In an effort to support NASA's goal to continue exploring the planet Mars, the JPL instituted MBSE by developing a second rover from the Mars Science Laboratory's (MSL) original design (Fosse, Devereaux, Harmon, & Lefland, 2015, p. 1). JPL's project provides many insights regarding the integration of legacy systems, modeling, ontologies, and the advantages and difficulties associated with MBSE. The JPL use-case authors state the following important lessons learned:

- Investment is crucial
- Unity of leadership is essential
- Best way to start modeling is to hire people who already know how to do it
- Team organization matters
- Everyone needs training, but not the same depth
- Best way to figure out how to apply MBSE: do it for real
- Keep the focus on project deliverables, and model only as far as you need to answer the questions

- Description first, then analysis
- Separate models from analyses
- Real examples are powerful (Elyse Fosse, 2016, pp. 72–73)

Many of these lessons learned apply to and ultimately assisted in the development of the CONOPS of the EE.

The second case is The Boeing Company's implementation of MBSE (Malone, Friedland, Herrold, & Fogarty, 2016). Much like the DoD, The Boeing Company must service a diverse array of product domains and, therefore, utilizes incremental approaches to develop and implement MBSE capabilities across legacy programs and new programs. These approaches, in addition to the identification of a wide spectrum of challenges, present many learning points to include in the CONOPS for the EE in the DoD. Organizational lessons learned offered from The Boeing Company include:

- System architecture models [are] indispensable at Boeing;
- High fidelity modeling allows Boeing to accelerate development schedules;
- Import and export utilities are critical;
- The dataset is the model (Model sharing is dataset sharing);
- Need several model views to efficiently populate and review data;
- Model analysis utilities are critical (Query Engine);
- A standard modeling notation does not achieve data integrity (a standard data model constrained by rules achieves integrity);
- Large model datasets bring data management challenges. (Malone et al., 2016, pp. 20–23)

2. Models

Models are the basic element of the EE that allow follow-on processes to inherit data. Models are "used to capture the system's behavior, structure, constraints, interfaces, and requirements" (Malone et al., 2016, p. 8). The use of models in systems engineering is not a new practice, but the use of recently developed standard formal modeling languages significantly improves the effectiveness and usefulness of current models. The EE leverages system modeling language (SysML) and archetype modeling language (AML) to generate complex, formal models. The formal modeling process requires many hours of operator interviews and the modeling of doctrine and tactics, techniques and procedures (TTPs), making it neither simple nor trivial (S. Miller, personal communication, March 27, 2017). This time spent on model development reduces problems identified in testing and affords significant savings in time and money in the long run, as Figure 2 demonstrates using The Boeing Company use case (Malone et al., 2016).

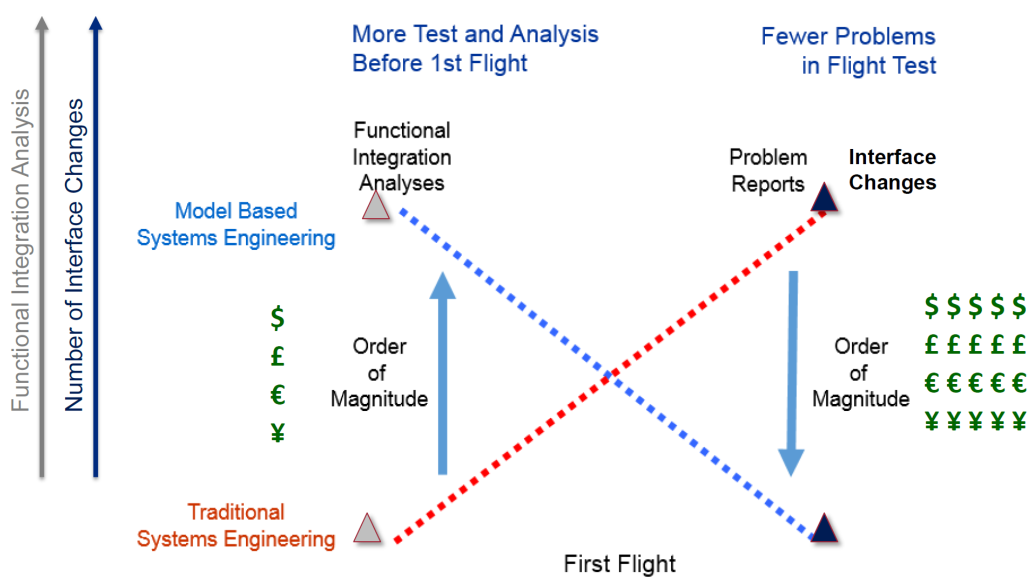


Figure 2. MBSE versus Traditional Systems Engineering Comparison for The Boeing Company. Source: Malone et al. (2016).

Formal models are the authoritative information source throughout all phases of design, implementation, and life cycle (S. Miller, personal communication, March 27, 2017). Although the incorporation of MBSE is important to the EE, the key aspect of the EE is the formal model itself because it captures requirements, implements constraints, and facilitates the follow-on processes within the EE (N. Eaglestone, personal communication, October 11, 2017). The end state of the formal model provides consistency and continuity throughout the entire life cycle of the item and is ultimately comprehensible to both humans and computers (Elyse Fosse, 2016, p. 32).

Many engineers use models to help them with systems engineering. Very few engineers create formal models, since in the short term they tend to take more time to correctly develop. Indeed, if one was to just build a one-off application that never needed to be upgraded or connected to another system, then formal modeling would be a waste of time. However, formal modeling enables faster test and evaluation, supports parametric comparisons and simulations, allows for automatic code generation, and simplifies functional changes to the software, to name just a few of the benefits (N. Eaglestone, interview with author, October 11, 2017).

3. Parametric Modeling

The EE's incorporation of parametric modeling using SysML enables model testing through simulations to prove whether a concept or model will work (N. Eaglestone, personal communication, Oct 9, 2017). Parametric models accomplish this proof of concept by implementing model constraints inside the system and evaluating them using appropriate analysis tools (Friedenthal et al., 2008, p. 149). These tools test model integration within the system of systems (SoS) with the end-state of achieving interoperability between systems (Elyse Fosse, 2016, p. 46). Parametric modeling and simulation determine design feasibility by ensuring that components properly integrate and communicate in the SoS and, therefore, are integral and invaluable (Rainey & Tolk, 2015, p. 76). Figure 3 diagrams a simple conceptualization of the use of simulations within the SoS. Parametric modeling and simulation provides proper prior-planning preventing poor performance.

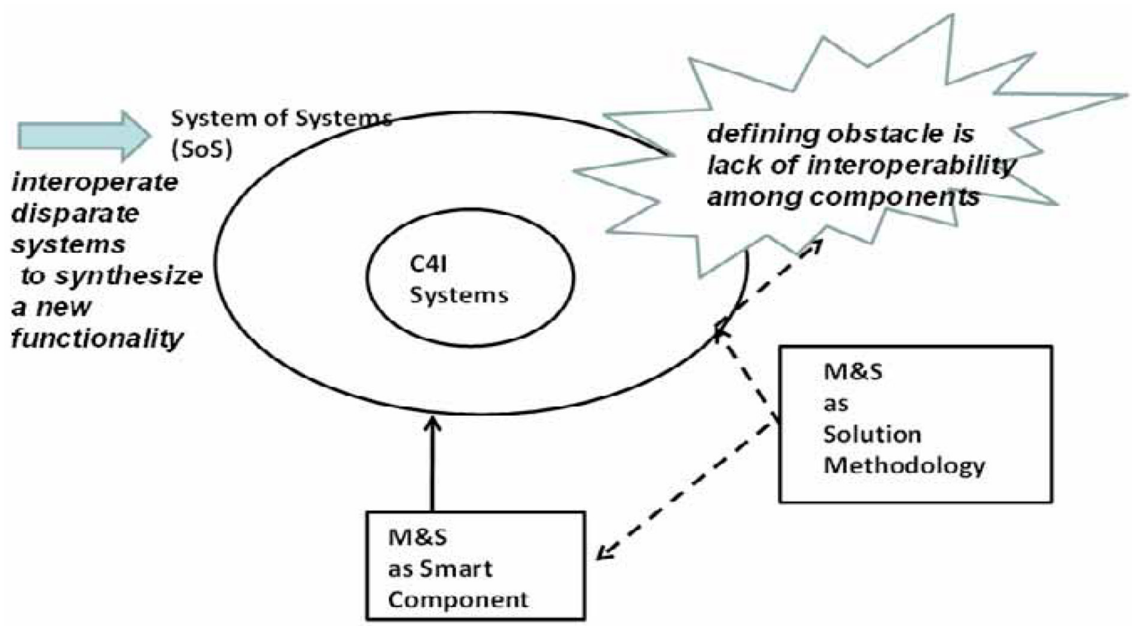


Figure 3. M&S as Source of Smart Components in C4I Systems.
Source: Mittal, Technologies, Zeigler, and Risco-Martin (2009).

Parametric modeling and simulations also facilitate trade space exploration (TSE) in order to conduct analysis of alternatives (AoA). TSE conducts this analysis of alternatives in order to determine cost and benefit trade-offs when new components enter the SoS (Rainey & Tolk, 2015, p. 75). The use of TSE within engineering has increased significantly over time. TSE across a SoS is considerably more difficult than in a single platform, but remains possible due to the use of formal modeling (Rainey & Tolk, 2015, p. 76). The diagram in Figure 4 demonstrates the addition of new systems into the SoS.

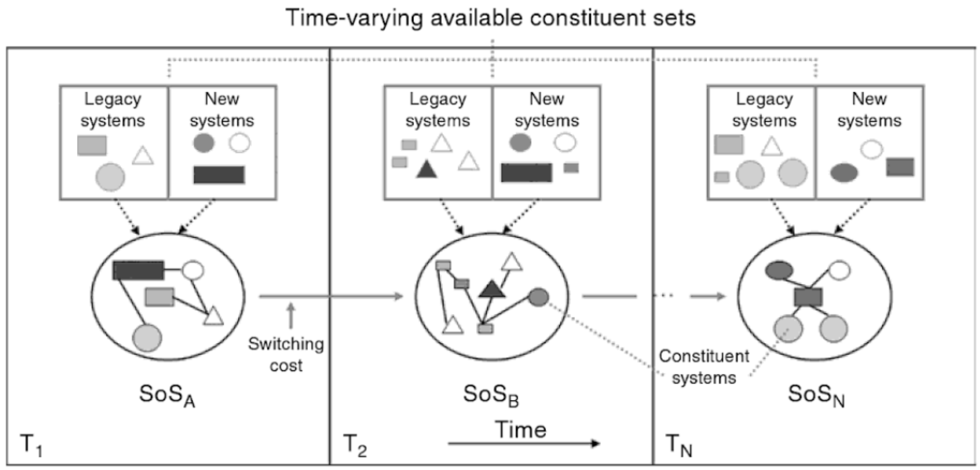


Figure 4. Legacy and New Constituent Systems in Time-Varying SoS Composition. Source: Rainey and Tolk (2015).

Parametric modeling and simulations provide the ability to conduct feasibility and interoperability tests throughout the life cycles of components within a SoS to ensure compatibility across the SoS. Results from these tests indicate whether it is necessary to make changes to the formal model before introducing it into the system or to a formal model already within the SoS. The ability to identify problems throughout the early stages of development ultimately saves time and money in the long run after the system formally enters into the SoS.

4. Standards

Standards are an important aspect of EE processes that enable EE to leverage industry-standardized processes and accepted practices. The EE will be built primarily upon open-source applications but will have the ability to interact with and include pre-existing proprietary programs and applications using developed endpoints and transforms. These transforms take advantage of previously identified patterns using variances to quickly allow new systems to integrate into the SoS.

Simply defined, standards improve quality by ensuring that development meets stakeholders' and industry requirements (Walden et al., 2015, p. 60). Modeling standards provide common foundations, enable model data connections and exchange, and allow for transformations to assist in semantic interoperability (Walden et al., 2015, p. 186). The benefits of adopting industry-wide standards are apparent in the example of the early development of electrical devices: the standardization of electrical plug sizes enabled devices to connect to the electrical grid, while the use of standardized power voltages ensured that devices operated within certain voltage ranges without being damaged. These standards allowed for multitudes of varying devices to integrate into the power grid and allowed innovators to focus on innovations and products rather than on interoperability and interchange properties (N. Eaglestone, interview with author, October 11, 2017).

There are many organizations that help develop standards. One such organization is the Object Management Group (OMG). OMG is a non-profit organization, the sole focus of which is developing industry-wide standards. The OMG is comprised of visionaries across all industries who believe in the benefits of standards for the good of all. OMG's mission statement is "to develop technology standards that provide real-world value for thousands of vertical industries. OMG is dedicated to bringing together its international membership of end-users, vendors, government agencies, universities and research institutions to develop and revise these standards as technologies change throughout the years" (Object Management Group, n.d.-a).

One of the keys to modeling standards is the use of common modeling languages that support model and data exchange (Friedenthal et al., 2008, p. 13). To that end, OMG utilizes common, standardized modeling languages such as the SysML, Unified Modeling Language (UML), Meta Object Facility (MOF) and AML. Appropriately, EE tools use these standardized languages and open-source applications in order to support the primary end states: rapid SoS integration, semantic interoperability between systems, and maximization of the use of automated code generation in development.

The use of standards also enables the integration of proprietary programs. Standards-based transforms and endpoints allow for these programs to integrate into the SoS by incorporating them into the ontology (N. Eaglestone, interview with author, October 11, 2017). Transforms take advantage of previously identified patterns using variances to quickly integrate new systems into the SoS. Subsequent sections of this chapter present further research on transforms and ontologies.

As EE matures, it may have the capability to institute internal standards (beyond OMG, Institute of Electrical and Electronics Engineers [IEEE], and other outside standards) to further ensure interoperability, decrease possible security gaps, and maximize the leveraging of the EE processes. Standards allow for the inclusion of rapidly developed commercial innovations while ensuring exclusive advantages and security (Object Management Group, n.d.-c).

5. Model-Driven Development

Standards drive the EE's development approach by allowing for transforms and fostering the compilation of ontologies. As this literature review contended at the outset, this will result in greater levels of interoperability, integration, and provisioning, while also maximizing the use of automated code generators to develop up to 16,000 lines of secure code per minute, compared to the two to four lines of code per minute of manual coding (N. Eaglestone, interview with author, October 11, 2017). EE's MDD platform relies upon multiple development tenets, all of which are crucial to realize the EE's full potential. Figure 5 displays these tenets.

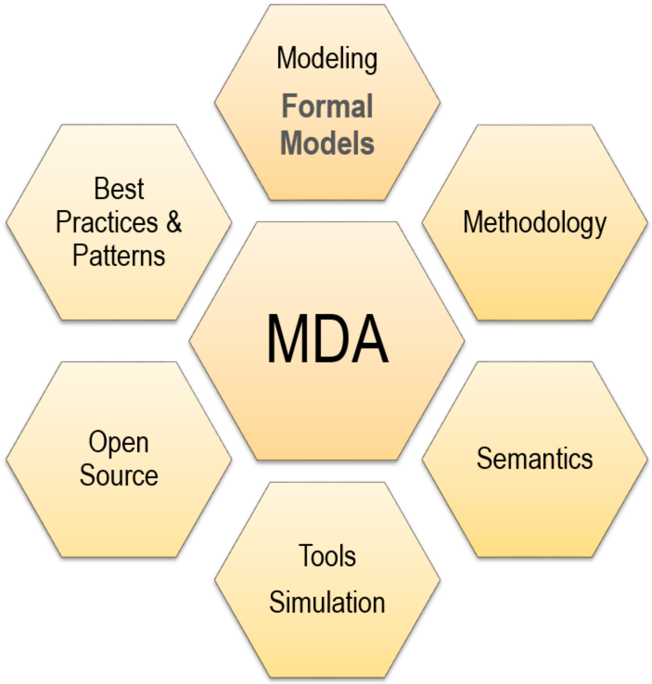


Figure 5. Model Driven Architecture Tenets. Source: N. Eaglestone, personal communication (2017).

The EE's MDD platform derives from the OMG's Model Driven Architecture (MDA), which OMG established as the base architecture for its standards in late 2001 (Object Management Group, n.d.-b). MDA's primary goal is to maximize the value extracted from models in order to address increased complexities and interdependencies in complicated systems (OMG, 2015, p. 2). MDA standards benefit development in three identified areas:

- [They provide] well defined terms, icons and notations that assist in a common understanding of a subject area
- [They provide] the foundation for models as semantic data to be managed, versioned and shared

- [They provide] libraries of reusable (asset) models such as common vocabularies and rules, reusable processes, business object models, or architectural design patterns (OMG, 2015, p. 2).

With MBSE and simulations providing the foundation for capturing requirements and standards, which enables commonality between platforms, EE's MDD platform takes those requirements and captures them in models in order to create computation independent models (CIMs), platform independent models (PIMs), and then finally platform specific models (PSMs) (Eaglestone, interview with author, 2017).

depicts this layered process. The development of a CIM provides a "view of a system from the computation independent viewpoint. A CIM does not show the details of the structure of systems" (Mukerji & Miller, 2003, p. 15). This CIM bridges the gap between subject-matter experts (SME) and modelers building formal models that must satisfy development requirements (Mukerji & Miller, 2003, p. 15). The CIM then develops into a PIM. This PIM provides a general system view independent of the platform, thus making it appropriate for use across multiple platforms (Mukerji & Miller, 2003, p. 16). This PIM contains the data that would satisfy Department of Defense Architecture Framework (DoDAF) requirements for DoD programs. OMG actually has tools that convert the PIM to those DoDAF views. The PSM then develops from the PIM. This PSM is from a platform-specific viewpoint of the system and incorporates platform-specific requirements into the PIM (Mukerji & Miller, 2003, p. 16).

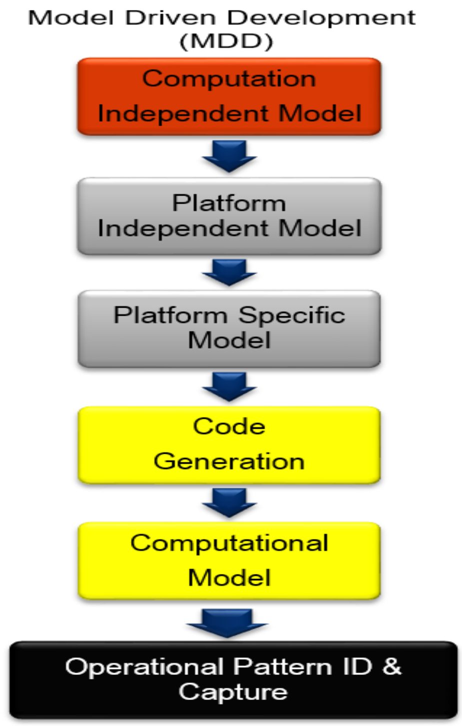


Figure 6. Model Driven Development Process. Source: N. Eaglestone, personal communication (2017).

a. Transforms

Key to MDD's success is the ability to perform transforms. The focus of these transforms should be on the overall SoS, not the individual platform (Eaglestone, interview with author, October 11, 2017). The development of PSMs and PIMs can help identify patterns that the transformation process then exploits. The EE currently plans on conducting at least three different types of transforms. The first will be the model-to-model transforms. The development of PIMs and PSMs require these transforms. Transforms will exploit variance and invariance to allow for the reuse of legacy models in the production of new models, which reduces development time dramatically. Figure 7 displays a visual depiction of the PIM to PSM transform process.

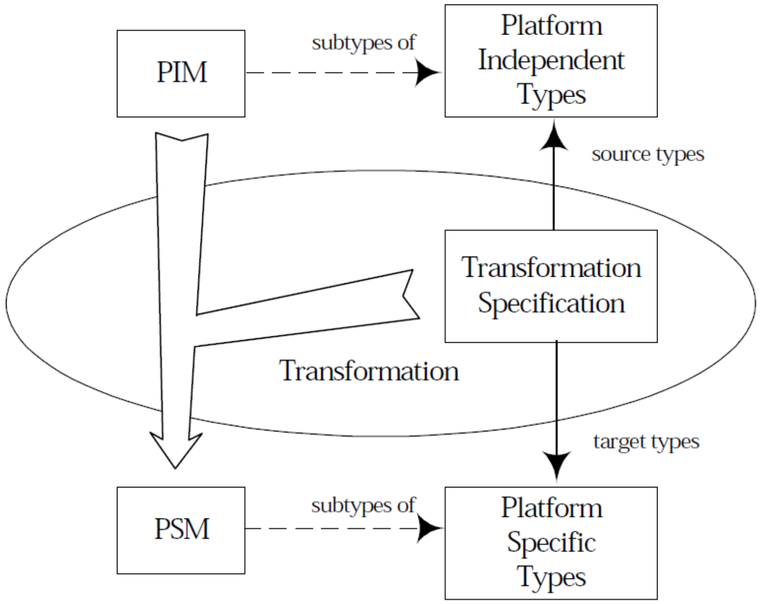


Figure 7. PIM to PSM Transform Diagram. Source: Mukerji and Miller (2003).

The second transforms type will be model-to-code transforms. Utilizing a centralized enterprise compiler, this process will transform PIMs into auto-generated pieces of code. MOF is key to transforming models into machine-readable data (OMG, 2015, p. 14). This process ultimately transitions the coding process from humans to machines and "allows humans to do what they do best: consider all options and employment considerations, understand the operating environment, and address constraints. It allows machines to do what they do best, which is to keep this information for future use and reuse, and to produce code at least 100,000 times faster than humans" (D. Boger et al., 2018, p. 6).

The final transforms type will be MOF-to-text transforms. These transforms generate requisite systems documents to support administration, acquisition, training, and development. This capability will reduce the amount of time that document generation requires. Due to the current multitude of document requirements within the development and acquisitions process, the importance of this transform cannot be overstated. Current development approaches lack the ability to keep system documents up-to-date. A formal model, if maintained properly, is a living thing, reflecting reality at any given moment. Conversely, as soon as a document prints, it becomes outdated. These MOF-to-text transforms enable users to always be able to print out the latest documentation with any changes to the system.

The open model approach espoused by EE's MDD enables the use of other future transforms not yet considered. This is the power of an open model approach, and it gives developers, program managers, and the acquisition community greater agility and responsiveness. This allows "systems to evolve, reuse, or integrate without substantial rework" (Elyse Fosse, 2016, p. 35).

B. EE INTEGRATION AND PROVISIONING APPROACH

The EE incorporates three primary foci in its integration and provisioning approach. The first is on SoS integration based upon the Levels of Conceptual Interoperability Model (LCIM). The LCIM provides a standardized spectrum that allows the quantifiable measurement of interoperability success. After establishing the LCIM integration approach, the next focus is on EE's tooling to ensure a successful integration. The EE's RCPI platform, which incorporates level-specific tools for technical, syntactic, and semantic integration, accomplishes this focus. EE's final focus is on provisioning tools used to address the pragmatic, dynamic, and conceptual levels of interoperability. The ultimate measure of success for the EE in its integration and provisioning approach is the transition from the small-data-era, Human-in-the-Loop Systems Provisioning to the big-data-era, Human-on-the-Loop SoS Provisioning (N. Eaglestone, personal communication, October 11, 2017).

1. Levels of Conceptual Interoperability Model

The first focus of the EE in its integration and provisioning approach is on SoS integration based upon the LCIM. The LCIM was established in 2003 to "become a bridge between the conceptual design and the technical design for implementation, integration, or federation" (D. A. Tolk & Muguira, 2003, p. 1). C.D. Turnitsa revised the LCIM in 2007 and Figure 8 presents that version.

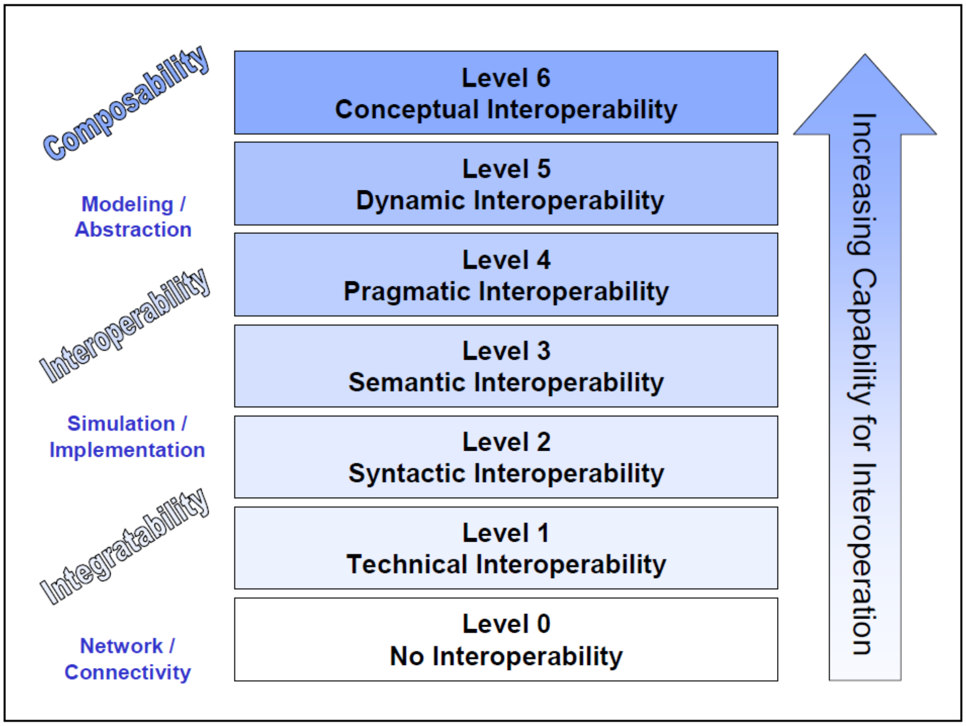


Figure 8. Levels of Conceptual Interoperability Model. Source: A. Tolk, Diallo, and Turnitsa (2007).

The EE's lead visionary, Norman Eaglestone, developed the EE's integration platform concept by taking each of the LCIM levels in account. Figure 9 describes each of the LCIM's levels and provides a short background narrative for each level.

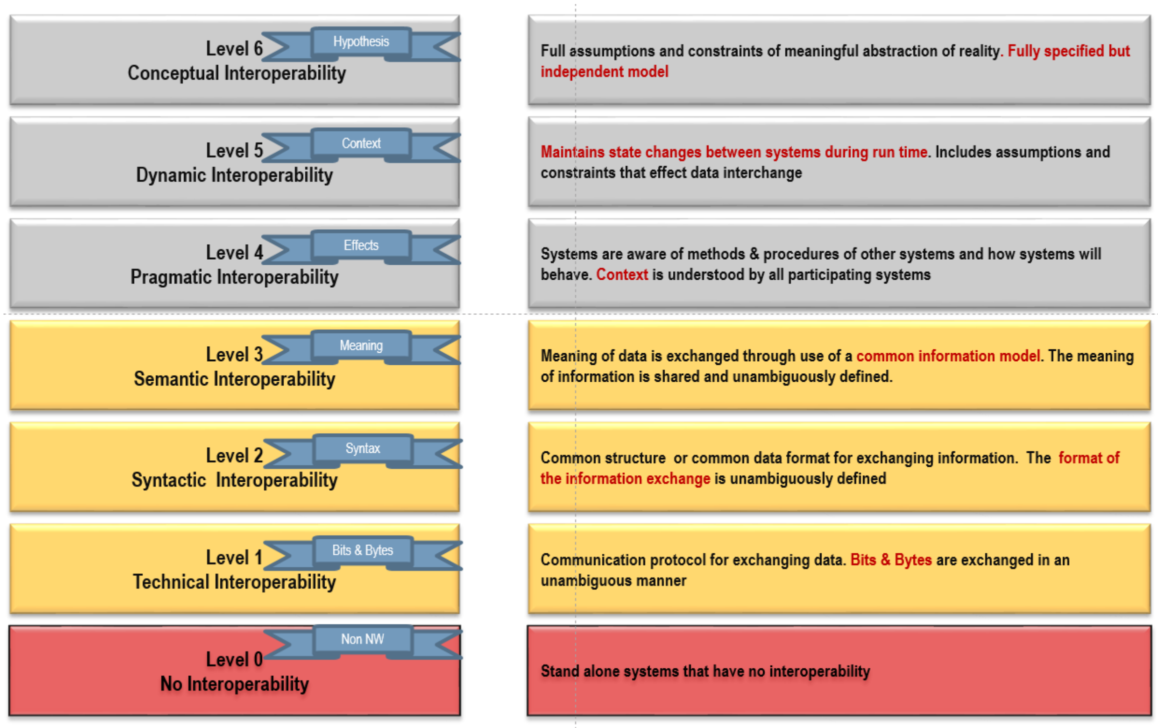


Figure 9. Expanded LCIM with Brief Description. Source: Eaglestone, personal communication (2017).

The fields depicted in yellow in Figure 9 are levels of interoperability on which the current development environment commonly focuses. The fields in gray depict levels that future development must expand to encompass. Transitioning to level 4 and above will most likely require modeling and simulations (N. Eaglestone, interview with author, October 11, 2017).

2. RCPI Platform

Successful SoS require independently built systems to operate on "common deoxyribonucleic acid (DNA)" in order to quickly allow and enable integration and interoperability (N. Eaglestone, interview with author, October 11, 2017). Interoperability will require more than application programing interfaces (APIs) to accomplish this interoperability. Complete SoS interoperability will require meeting all levels of interoperability, from the technical to the conceptual levels (N. Eaglestone, interview with author, October 11, 2017). The RCPI platform of the EE begins with addressing levels 1–

3 of interoperability because they are necessary for a new system to properly function upon introduction into a SoS. Future development of provisioning tools will further address the pragmatic, dynamic, and conceptual levels working in concert with a MDD toolkit. As operational tempo increases, the RCPI transforms the formal model that derives from the requirements development process into rapidly provisioned capabilities, ultimately resulting in the deployment of capabilities prior to reaching obsolescence (N. Eaglestone, personal communication, October 26, 2017).

a. EE Technical Integration

Technical interoperability (Level 1 on LCIM) in its simplest terms is the exchange of bits and bytes between systems. Communication infrastructure imparted with a series of communication protocols accomplishes this exchange across networks (A. Tolk et al., 2007, pp. 66–67). The EE's RCPI platform intends to enable technical integration within the SoS through the inclusion of end points during the modeling process. These end points are essentially the connectors between systems. Figure 10 is a visual depiction of endpoints that a demonstration of an adaptive detect, classify, and track mission uses, and presents the importance of the connections between systems.

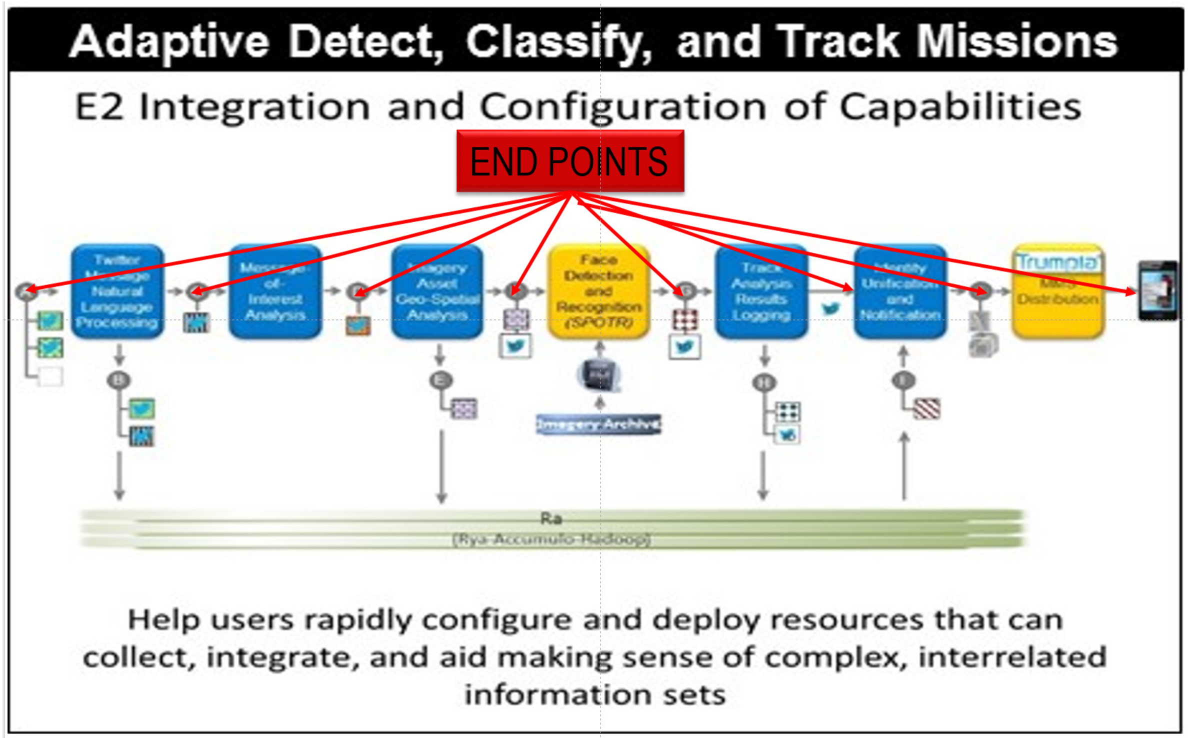


Figure 10. Visual Depiction of Usage of Endpoints. Source: N. Eaglestone, personal communication (2017).

Endpoints will also have the ability to enable storage of meta-data, which will result in the ability to reuse endpoints within a repository. The storage of endpoints in a repository of endpoints makes future technical integration much faster (D. Boger et al., 2018, p. 6). Endpoints are more complex and sophisticated in nature than traditional APIs. Systems that the EE develops with endpoints must adhere to a mandatory set of system requirements or contract, which results in significantly faster connections to the SoS (N. Eaglestone, interview with author, October 11, 2017). These requirements may include, but are not limited to, system-specific APIs, security requirements, and any other necessary components the SoS may require for adherence to the SoS contract. The SoS achieves technical interoperability once each system, installed with common endpoints, fully integrates to comprise it.

b. EE Syntactic Integration

Syntactic Interoperability occurs when common transmission structures effectively exchange information between systems (A. Tolk et al., 2007, p. 67). The early stages of systems development encountered many issues at the syntactic level because each system developed independently and, therefore, had its own distinct environment. As web services and transmissions matured, a common transmission structure with transmission protocols resolved many of these syntactic issues (N. Eaglestone, interview with author, October 11, 2017). Installing common web protocols and programming during model development will ensure the maintenance of syntactic interoperability across the SoS.

c. EE Semantic Integration

Achieving semantic interoperability (Level 3 on the LCIM), which ensures that systems share a common meaning of data, is a key goal of the EE's RCPI platform because it allows for rapid, cost-effective systems integration as new systems enter into the larger SoS (A. Tolk et al., 2007, p. 67). The ability to provide these reduced costs is noteworthy because systems-integration costs are directly related to the square of the number of systems to be integrated (D. Boger et al., 2016). A simple example that highlights the importance of semantic interoperability is a system's potential to misinterpret the words 'tank' and 'port'. These words both take on different meanings depending on the context of their use: information passed between systems pertaining to tanks could refer to water storage tanks or tank weapon systems, while ports could refer to seaports or ports on a computer. For a SoS to truly have semantic interoperability, all systems within the SoS must interpret the meaning of data uniformly, which formal models make possible. (N. Eaglestone, interview with author, October 11, 2017).

Semantic interoperability ensures that machines and humans seamlessly exchange contextual information between systems, which is a necessary step toward achieving man-machine collaboration and is central to achieving success in machine learning, deep learning, and other artificial intelligence techniques (D. Boger et al., 2016). This capability is made possible by using ontologies derived from formal model metadata and OMG's AML, born from efforts to integrate various health care systems (D. Boger et al., 2018, p.

6). These tools will enable semantic interoperability between integrated systems, which is a huge accomplishment (Boger et al., 2018, p. 6). Previously, only expensive and time-consuming programming that was inflexible and resistant to configuration changes could achieve this (Boger et al., 2018, p. 6). Achieving successful semantic interoperability will result in more adaptable SoS, which will quickly allow systems to change in response to the needs of the organization, develop ontologies for use at the higher levels of interoperability, and drive down costs while enhancing speed to capability (N. Eaglestone, interview with author, October 11, 2017)..

d. Pragmatic/Dynamic/Perceptual Integration

The LCIM's Levels 4–6 are the ultimate end state desired in interoperability. These levels include:

- Level 4 Pragmatic Interoperability: systems within SoS comprehend the context of information, methods, and procedures being used by partner systems;

- Level 5 Dynamic Interoperability: systems within the SoS have the ability to understand state changes occurring during operations within the system and then have the ability to capitalize on those changes;

- Level 6 Conceptual Interoperability: Assumptions and constraints are conceptually aligned and with all aspects and meanings being fully understood within the SoS (A. Tolk et al., 2007, p. 67).

The EE's RCPI platform intends on progressing toward higher levels of interoperability (LCIM Levels 4–6) within the SoS. This provisioning platform will use provisioning tools to exploit the depth of information that the formal models and associated ontologies contain. The development process includes the collection of provisioning tools, warranting the name Model Driven Development Toolkit (MDK), which Figure 11 depicts (N. Eaglestone, personal communication, October 11, 2017).

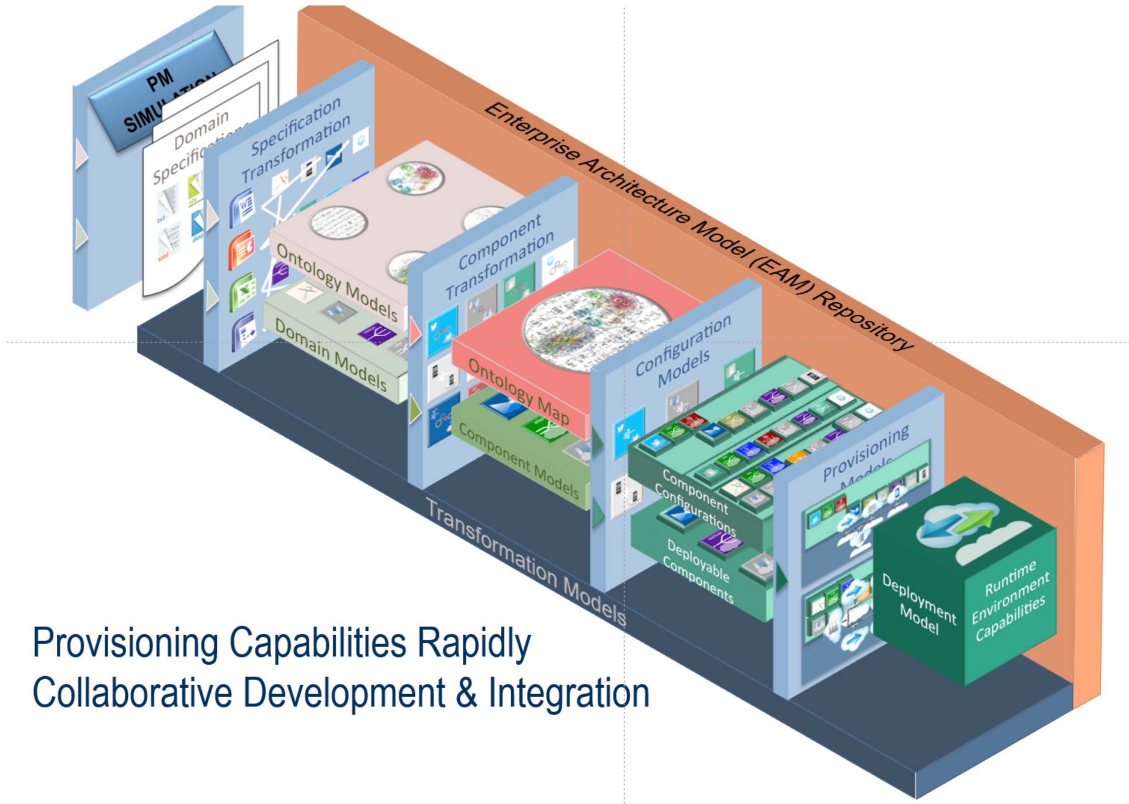


Figure 11. Model Driven Development Toolkit (MDK). Source: N. Eaglestone, personal communication (2017).

Progressing toward LCIM Levels 4–6 using MDK's tools capitalizes on generating assets from store, executing necessary transformations, and instilling the necessary contextual and conceptual understanding across the SoS.

C. EE SECURITY APPROACH

Information assurance (IA) and adherence to the Risk Management Framework (RMF) to combat cybersecurity hazards for information systems is increasingly important in the DoD. The ability to provide continuous cyber monitoring combined with the ability to leverage the MDK and RCPI tools allows the EE to address cyber threats and overcome them quickly. The Model Oriented Development Environment (MODE) is the first of the EE-developed tools that begins to address these issues. Figure 12 diagrams the composition and initial capabilities.

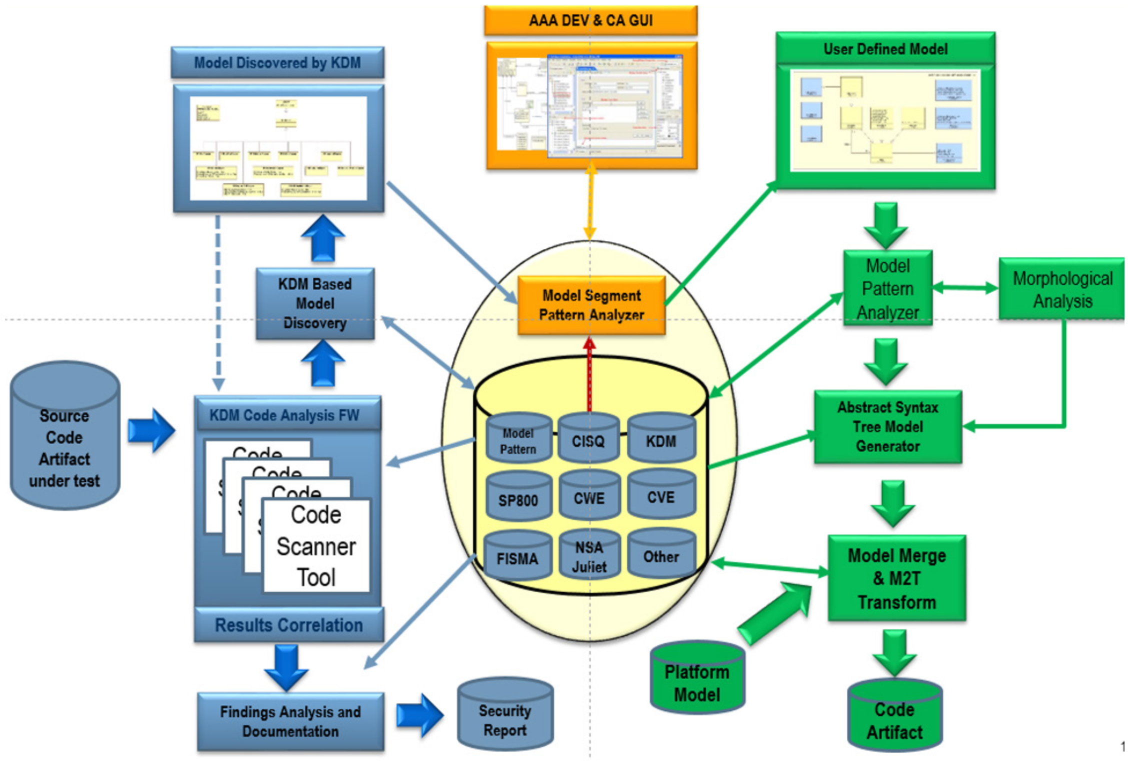


Figure 12. MODE Conceptual Core Architecture. Source: M. Koethe, personal communication (2017).

Accelerating the Risk Management Framework (Kim, 2018) details the EE's and MODE's security processes as well as the positive impacts on information assurance and the RMF process. Kim (2018) summarizes the impacts of EE and MODE thus:

> With the advent of semantic interoperability and enhanced MODE capabilities, the assessment of capabilities rapidly integrated on the fly to meet emergent operational needs will be able to be performed in days or hours, instead of the current months or years. The contribution to the DoD's ability to be adaptive will be immense (pp. 67–68).

D. CURRENT ACQUISITIONS PROCESS

Developing a CONOPS for integrating the EE into a SoS requires an understanding of the current Integrated Defense Acquisition, Technology, and Logistics Life Cycle Management System. This system is very complex and comprises three different decision-making support systems: the Joint Capabilities Integration and Development System (JCIDS), the DAS, and the Planning, Programming, Budgeting, and Execution (PPB&E)

process. The chart in the Appendix, which acquisition professionals commonly refer to as the acquisitions system wall chart, portrays generic synchronization of the internal processes of the three systems. Although different approaches drive these three systems, they all converge to ultimately provide capabilities that support the warfighter. The DAS is how we acquire systems and is events-driven (W. Fast, class notes, July 12, 2017). The JCIDS determines requisite capabilities and is needs-driven (W. Fast, class notes, July 12, 2017). Finally, the PPB&E allocates resources and is calendar-driven (W. Fast, class notes, July 12, 2017). The President of the United States' (POTUS) strategic guidance subsumes each of these systems, which Figure 13 depicts, while all three conjoin with the Quadrennial Defense Review (QDR), which is the "single hierarchical link integrating all internal decision processes" that Figure 14 illustrates (W. Fast, class notes, July 12, 2017).

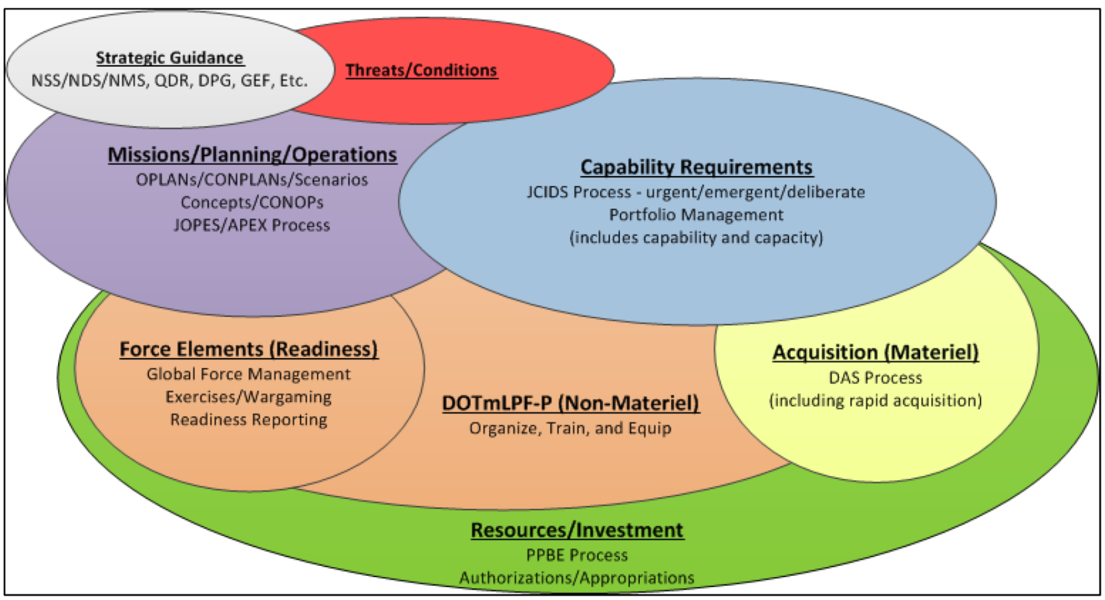


Figure 13. Process Interactions. Source: Department of Defense (2015).

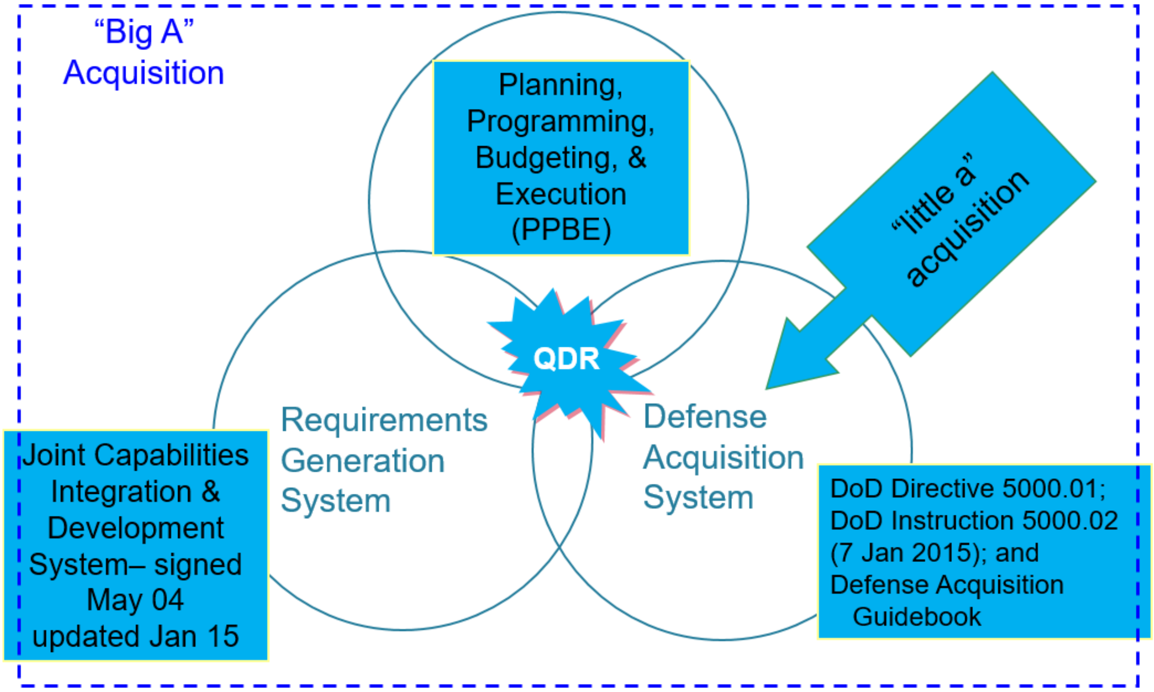


Figure 14. Decision-making Support System Linkage. Source: W. Fast, class notes (2017).

Although the PPB&E process is an important aspect of acquisitions, it is outside the scope of this thesis. Rather, the thesis will present a basic introduction of the JCIDS process (especially material and non-material solutions) and explore the EE's role as a possible enabling solution for a more adaptable and efficient DAS process.

1. Joint Capabilities Integration and Development System

The JCIDS is a top-down requirements generation process that the Joint Chiefs of Staff drive and oversee. The POTUS' strategic guidance steers JCIDS analyses. With this guidance in mind, the Joint Chiefs use the JCIDS process to conduct a Capabilities Based Assessment to determine capability gaps and identify possible solutions to fill these gaps. Figure 15 provides a flow diagram of this process that uses either a material solution (DAS) or non-material solution (Doctrine, Organization, Training, Material, Leadership, Personnel, and Facilities [DOTMLPF]) to fill support gaps.

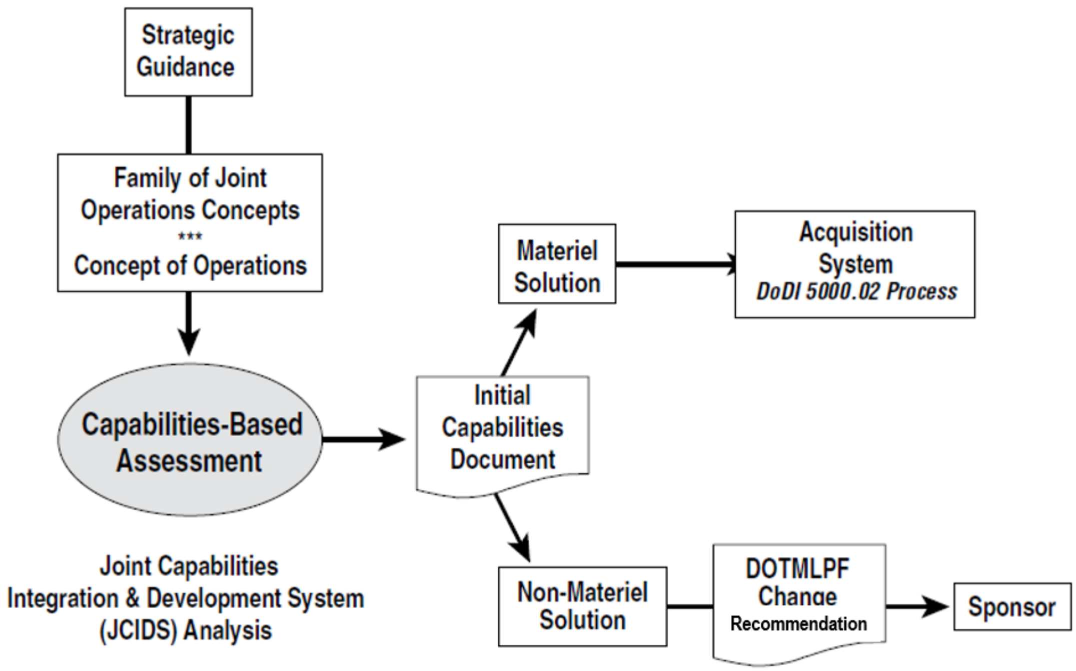


Figure 15. JCIDS Capabilities Process. Source: W. Fast, class notes (2017).

The JCIDS process generates three important documents that integrate into DAS milestones in order to effectively address capability requirements and associated gaps. The first important JCIDS document is the Initial Capabilities Document (ICD) that the Material Development Decision in the DAS process requires. The next DAS process milestone (B) requires the Capability Development Document (CDD); Milestone C then requires the Capability Production Document (CPD). Within the CDD and CPD are key performance parameters (KPP), which are "performance attributes considered critical to the development of an effective military capability" (W. Fast, class notes, July 12, 2017). Figure 16 outlines each of these documents with the requisite "trade space" between threshold and objective values (W. Fast, class notes, July 12, 2017).

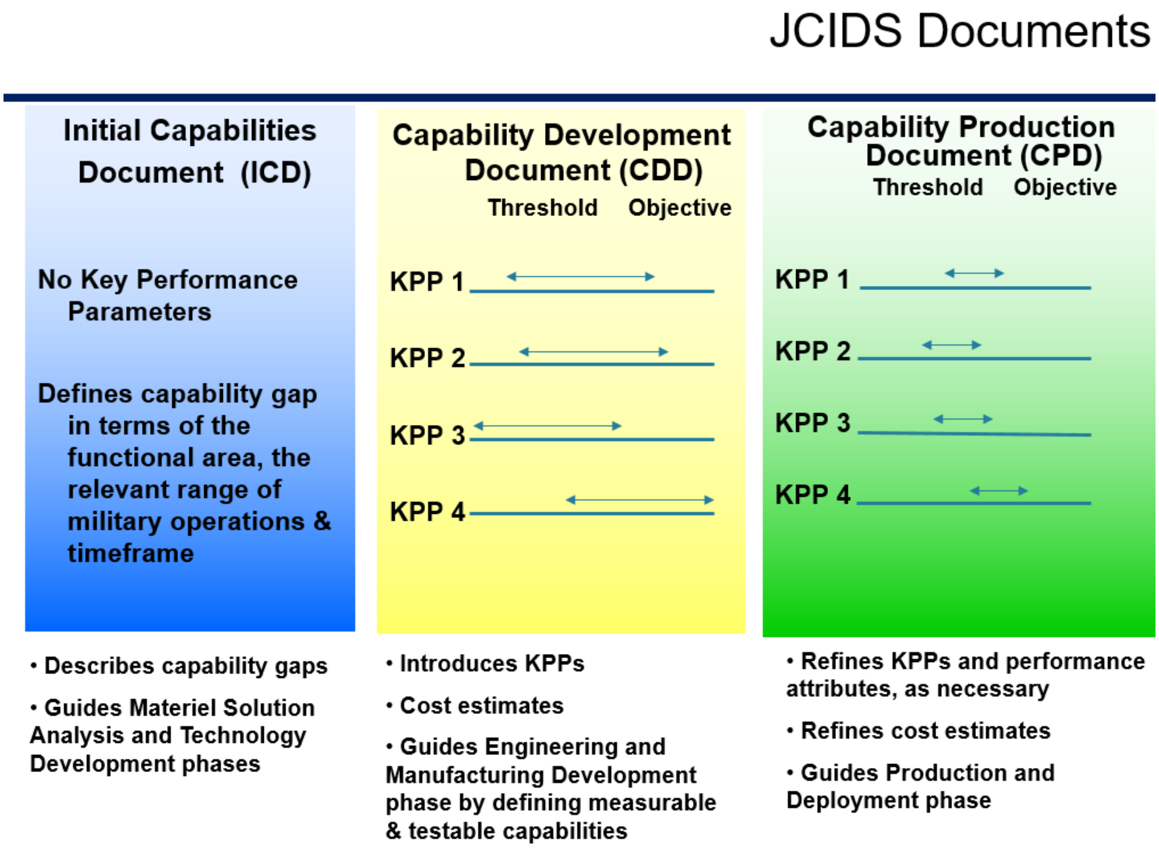


Figure 16. JCIDS Documents Description. Source: W. Fast, class notes (2017).

2. Defense Acquisition System

As JCIDS identifies needs, the DAS inherits the material-based solutions requirement and determines how to acquire them. As previously identified, DAS is a phased events-driven process that "progressively develops, produces, and fields useable materiel to meet the capability needs of the warfighter" (W. Fast, class notes, July 12, 2017). Figure 17 encapsulates the DAS phases and milestones with the applicable JCIDS requirements.

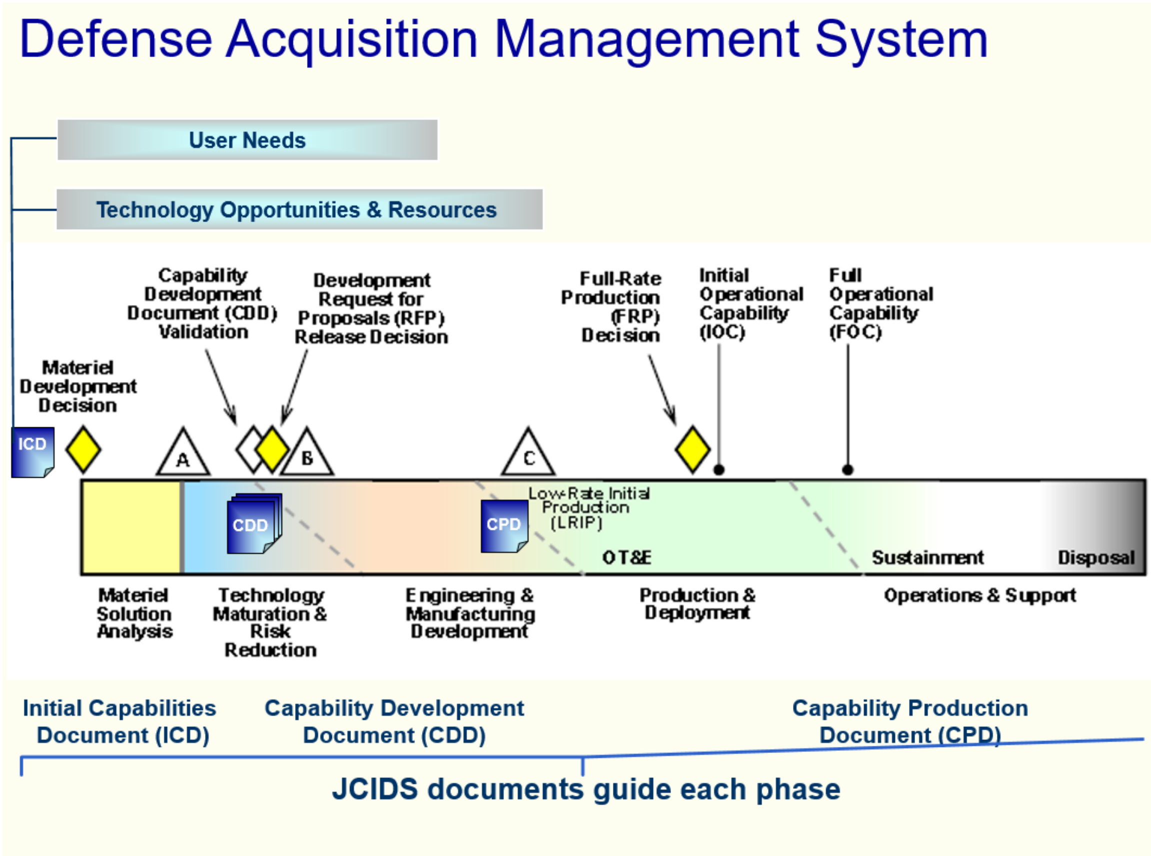


Figure 17. DAS and JCIDS Phase Diagram. Source: W. Fast, class notes (2017).

Beyond the basics of the DAS process, incremental development and modular open systems approaches (MOSA) are key DAS requirements that the EE's development and integration approaches could greatly enhance and support. MOSA employs modular designs and emphasizes the use of consensus-based standardized key interfaces recognized by industrial standards organizations (W. Fast, class notes, July 12, 2017). Both the DoDD 5000.01 and DoDI 5000.02 outline the MOSA requirement. Figure 18 depicts the general incorporation of incremental and MOSA developments.

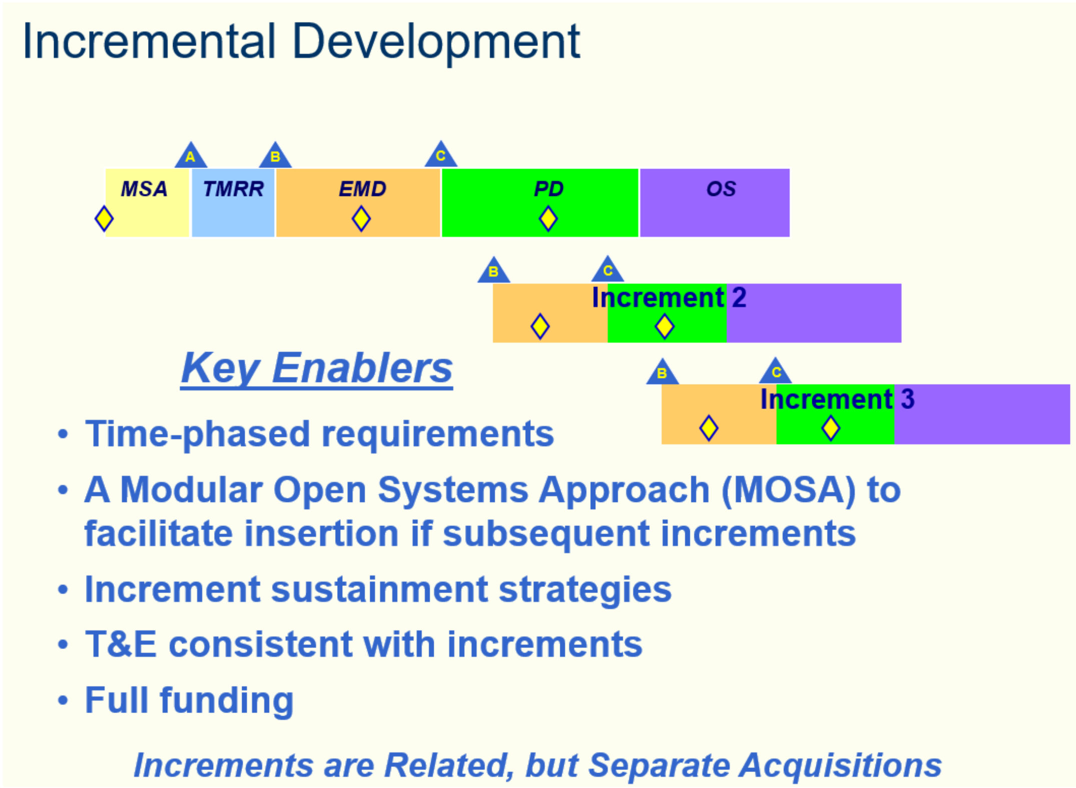


Figure 18. Incremental Development Diagram. Source: W. Fast, class notes (2017).

E. BASICS OF CONCEPT OF OPERATIONS

The development of a CONOPS is of paramount importance. IEEE defines CONOPS "as a user oriented document that describes system characteristics of the to-be-delivered system from the user's viewpoint" ("IEEE Guide for Information Technology - System Definition - Concept of Operations (ConOps) Document," 1998, p. 2). IEEE (1998) continues, describing the CONOPS as a "document used to communicate overall quantitative and qualitative system characteristics to the user, buyer, developer, and other organizational elements (e.g., training, facilities, staffing, and maintenance). It describes the user organization(s), mission(s), and organizational objectives from an integrated systems point of view" (p. 1). Although studies reveal that perceptions of CONOPS are that they are critical and underutilized, many programs continue to either lack a formal CONOPS or generate CONOPS after the development of requirements (Edson & Frittman, 2012, p. 7). Traceability and continuity are two values that can derive from integrated

CONOPS (Edson & Frittman, 2012, p. 14). Research has shown CONOPS' specificity increases as systems mature throughout three distinct phases: initial, discovery, and employment (Edson & Frittman, 2012, p. 16). Figure 19 illustrates this theory.

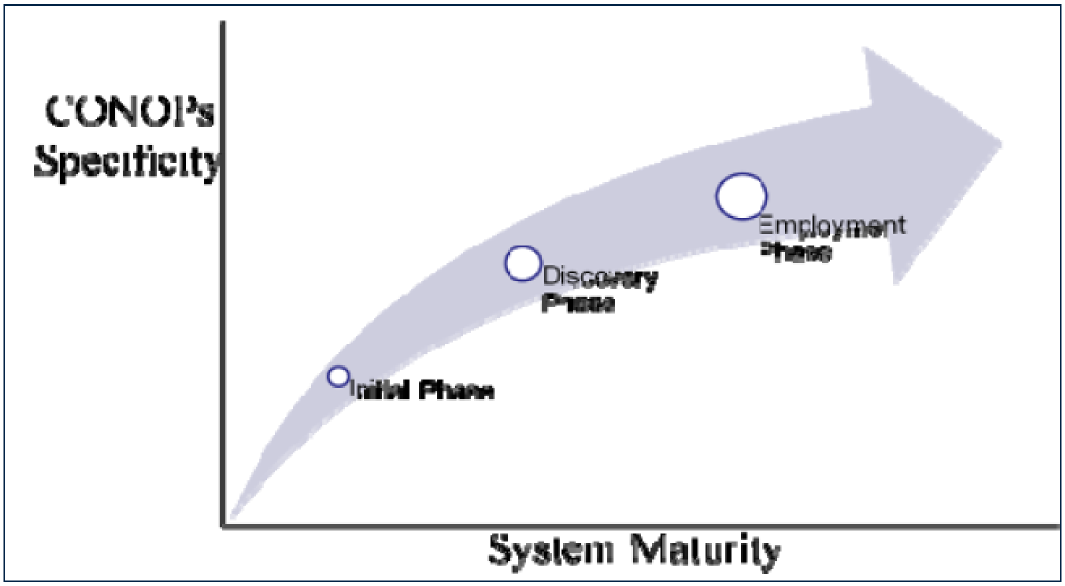


Figure 19. CONOPS Maturity Phases. Source: Edson and Frittman (2012).

With this theory in mind, researchers proceeded to align these CONOPS maturity phases to the DAS process, which Figure 20 depicts.

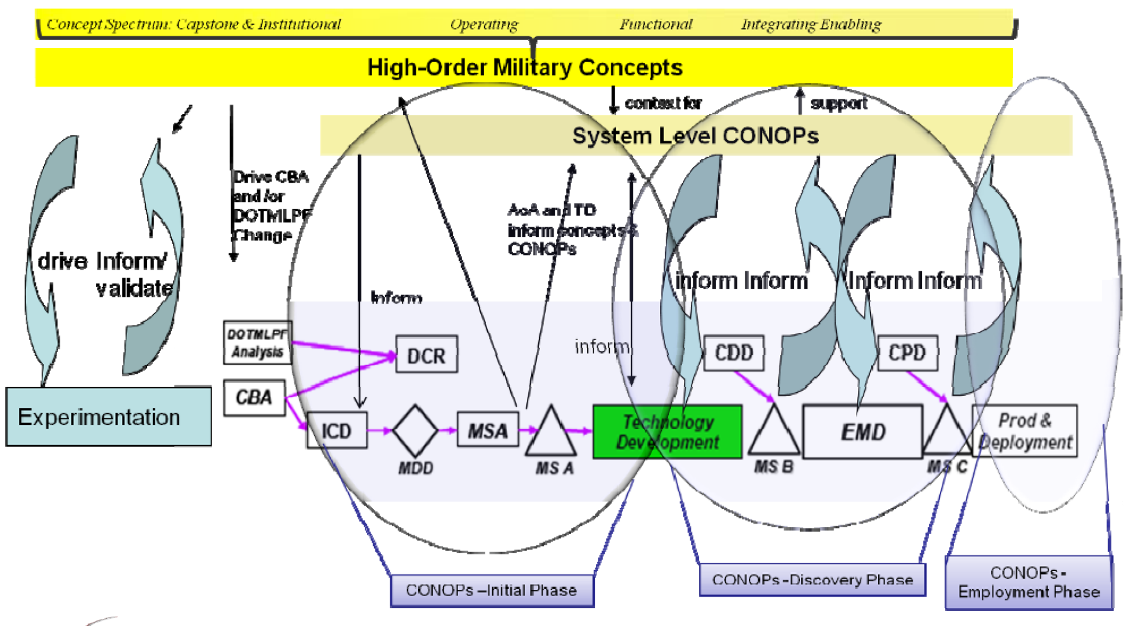


Figure 20. Alignment of Integrated CONOPS in the DAS. Source: Edson and Frittman (2012).

Due to being within the Initial phase, the CONOPS that this thesis proposes for the EE will be in its most ideal form, geared toward developing ICD requirements, and will mature as EE matures (Edson & Frittman, 2012, p. 17,19).

F. CHAPTER CONCLUSION

This chapter presented research that demonstrates the applicability of the fundamentals behind EE's developmental, integration, and security approaches and the future value they represent if leveraged in the future. It also contained necessary research to provide a basic understanding of the overall acquisitions process as well as stressed the importance of newly developed systems having initial CONOPS with a description of how these CONOPS will progress through maturity phases in relation to the DAS. This background research provides the fundamentals that will be instrumental to understanding the complexities and importance of deriving a proposed CONOPS for implementation of the EE.

THIS PAGE INTENTIONALLY LEFT BLANK

III. RESEARCH METHODOLOGY

This chapter outlines the research methodology. The purpose of this research is to develop a CONOPS for the EE that recommends implementation and guides EE users through its processes. Executing this research meant learning about acquisition and associated processes, understanding EE tools, and mapping the EE to various components of the acquisition process. Qualitative research methods were the primary means of research for examining the key components of the EE, surveying the current as-is military acquisitions environment, and finally developing a feasible CONOPS for the employment of the EE.

A. STUDY DESIGN AND IMPLEMENTATION

The qualitative methods that developed and accomplished our research included comprehensive examination of the important concepts and tools within the scope of employment of the EE. These areas included MBSE, MDD, ontologies, information security, and the DAS, along with other requisite interrelated topics.

The EE research team implemented various forms of research throughout the development of the CONOPS, spanning from December 2016 to September 2018, including:

- Classical classroom study via lectures conducted on DAS, Enterprise Architecture, Organizational Change, Enterprise Information Systems Strategy and Policy, and Project Management for Enterprise Systems (MN3331, CC4250, IS4182, IS4300). This instruction provided a baseline of knowledge on the current acquisition environment, successful organizational change approaches, and current DoD and commercial enterprise architecture design methods. The insights from these class lectures provided the necessary foundational knowledge for understanding the problem and for developing the CONOPS for the EE to possibly solve it.

- Interviews and conversations with subject matter experts in the fields of MBSE, EE, and the DAS. These interactions included the lead program manager of a current experimental Navy program using MBSE in development, the leading expert and originator of EE, the MODE lead engineer, and several representatives from multiple military systems commands and military programs offices including Naval Air Systems Command (NAVAIR), Naval Sea Systems Command (NAVSEA), Marine Corps Systems Command (MARCORSYSCOM), Marine Corps Warfighting Laboratory (MCWL), and USMC Programs and Resources (P&R).

- Attendance, participation, and presentation at the Acquisitions Research Symposium in Monterey, California. This annual three-day symposium attracts participants from throughout the commercial and defense industries, DoD program management offices, and DoD acquisition policy makers. During this symposium, members of the EE research team submitted a research-driven point paper and conducted a formal presentation on the EE during the "Applying Model Based Systems Engineering to Defense Acquisition" discussion panel. Participation in this panel resulted in valuable feedback regarding the EE concept from outside acquisitions and systems engineering SMEs as well as lessons learned from fellow presenters.

- Case-study analysis of past EE-related programs (Notice to Mariners [NtM], MODE, Unmanned Tactical Autonomous Control and Collaboration [UTACC]) that implemented EE tools, as well as the case studies related to past MBSE implementation in a high-profile individual space program and across a large, industry-leading company. These valuable case studies outline the implementation and use of MBSE for complex programs, reinforce the advantages associated with its use, and provide insight and lessons learned that the CONOPS for the EE can use.

B. EE PROCESS ANALYSIS

The research methodology that the study design describes established a deep understanding of the positive impacts that the implementation of the EE could have. The knowledge gained from each facet of research allowed for the development of a CONOPS for the implementation of EE. The proposed CONOPS for the EE attempts to identify and improve upon areas within the development, integration, and sustainment life cycles.

As of the writing of this CONOPS, some of the tools it describes are currently concepts envisioned by EE's developer but have not been developed. This means that the implementation of those capabilities will require assumptions until they are developed. In these cases, research on these assumed capabilities determined the feasibility of including them into the CONOPS of EE. Chapter II provided this outline of the feasibility research, Chapter IV will identify these assumptions in the description of the CONOPS and determine their feasibility and applicability, and Chapter V provides future recommendations to further advance the development of these tools.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ANALYSIS OF RESULTS AND OBSERVATIONS

A CONOPS simply explains how to use a new system or process. In this case, the EE is both a set of systems, tools and standards, and a developmental process. A CONOPS should cover the details of how all this works, and this chapter will describe those details. This CONOPS applies to users who are developing new systems, as well as to users trying to integrate new and legacy systems. The end of this chapter presents an example of how to use the CONOPS.

A. CONOPS SUMMARY

1. EE CONOPS Overview

This EE CONOPS outlines the current operational and developmental environment, identifies current capability gaps, and how the EE could remedy them. It also identifies the possible impacts of the EE and rules out doctrine, organization, training, leadership and education, personnel, and facilities (DOTLPF) changes as a solution to capability gaps. Finally, this CONOPS proposes potential JCIDS and DAS requirements by phase that the EE would benefit through the use of implementation scenario maps.

2. Proposed Capability

This CONOPS proposes the EE as a potential enterprise-wide life-cycle process that could potentially revolutionize the operational and developmental environments within the DoD. The EE consists of six main components—formal modeling via MDD, semantic enabling, automated code generation, code quality & cyber-health, rapid integration and provisioning (via the RCPI module), and deployment—that have the potential to address multiple capability gaps that exist in the current environment. The EE will allow the DoD to shift toward the use of formal models to combat existing cognitive overload encountered during the development of complex SoS, to become more operationally adaptive, and to rapidly tackle potential cybersecurity and risk management issues in an increasingly contested cyber environment.

B. CURRENT SITUATION AND CAPABILITY NEEDS

Operational and development environments that present the DoD with increasing and demanding challenges drive the current situational and capability needs. The current operational environment faces increased pressure to innovate due to emerging potential near-peer adversaries. This innovation must be accomplished across a dispersed documents-based systems development environment. Developing the necessary system of systems is increasingly more difficult since engineers encounter added complexity and integration requirements. Moreover, the current cyber environment requires that systems address increased security risks and vulnerabilities.

1. Current Policies and Constraints

The current operational and development environment has multiple policies with associated requirements and constraints, the adherence to which are mandatory. While there are too many to outline, four major policies that levy requirements in the current operational and development process are: JCIDS, DAS, DoDAF, and RMF. First, all systems development requires the use of JCIDS and DAS. These policies outline the requirements for managing development and acquisitions within the DoD. Next, the DoDAF is requisite for all information architectures in order to enable a "framework and conceptual model enabling development of architectures" (Department of Defense, n.d.). The DoD Chief Information Officer (CIO) requires adherence to DoDAF. Finally, all information technology systems must adhere to the DoD RMF in order to address security and cyber risks associated with the current environment. All of these policies, in addition to many not listed (weapons certifications, CYBER SAFE, etc.), add significant complexity in an already complex operational and development environment.

2. Support Environment

Within the current DoD operational and development environment, there are many organizations that exist to support the development of new systems. These organizations include service-level systems development commands and service-level innovation and warfighting laboratories. The purpose of these organizations is to focus on the development of future systems to ultimately support the warfighter. Necessary to the support environment are corporate defense systems developers who are contracted to fulfill development roles that cannot be filled organically by the DoD. All of these organizations providing development support encounter the challenges and capability gaps within the current environment.

Service-level systems commands assume the primary responsibility of designing, constructing, and providing maintenance of military systems. Each service task organizes systems commands differently, in a manner conducive to best support their service. The Navy systems development is split into five separate organizations: NAVSEA, NAVAIR, Space and Naval Warfare Systems Command (SPAWAR), Naval Facilities Engineering Command (NAVFAC), and Naval Supply Systems Command (NAVSUP). The MARCORSYSCOM, organized with eight primary subordinate commands divided primarily by function, assumes this responsibility for the Marine Corps. The Army Materiel Command has primary responsibility for the development of Army systems, and the Air Force Material Command is responsible for Air Force systems development.

Each service within DoD maintains research and innovation organizations to spur innovation and speed development. Key organizations within the Navy and Marine Corps are the Office of Naval Research (ONR) and the MCWL. The Air Force Office of Transformational Innovation and newly formed Army Futures Command will drive future developmental programs in those services. While these organizations are focused on development within their services, program development efforts are likely disjointed between each of the services.

Defense contractors are also important organizations within the current support environment. Some of the top contractors providing support are: Lockheed Martin Corporation, The Boeing Company, Raytheon Company, General Dynamics Corporation, and Northrop Grumman Corporation. Defense contractors assist in development in almost every acquisition program within DoD.

3. Capability Gaps

There are many capability gaps within the current operating and development environment that EE could remedy. Table 1 displays these gaps and their associated impacts.

Table 1. Capability Gaps with Environmental Impacts

Capability Gaps	Environment Impacts
Outdated Document Based systems development process	Systems paperwork immediately outdated; Systems are slow to adapt to necessary changes
Software Development Process	Delayed delivery of requisite capabilities due to code generation by humans and lack of ability to reuse code
Ability to integrate systems in SoS	Integration is slow and expensive for new integration of systems.
Lack of ability to maximize the re-use of system architectures and code due to information silos	Software code is developed from scratch for new systems, delaying the software process
Lack of Semantic Interoperability within SoS	Limits use of AI, machine-learning, deep-learning capabilities

C. CONCEPTS FOR THE PROPOSED SYSTEM

The EE is the product of a group of information systems and software development experts who believe the current development and acquisitions process must improve. This group envisioned a process using formal models that "will reduce costs, accelerate delivery, and improve operational performance" (D. Boger et al., 2018, p. 9). Current EE tools have demonstrated effectiveness in six proofs of concept. Table 2 displays these six proofs of concept with general capabilities and integration background information. This CONOPS will further describe two of these proofs of concept.

Table 2. EE Formal Model Proofs of Concept. Source: Boger et al. (2018).

Formal Model Proofs of Concept, 2012–2018				
Project Title/Date	General capability	Systems Integrated	Time	Cost
Counter battery 2012	Parametric modeling of sensors and networks	Four models	Three man months	$100k
Social network analysis 2014	Sensors to computer vision to tweet based alert network	Two sensors, facial recognition software, data bases, basic semantic interoperability; network integration; alert development	Three man months	$150k
Unmanned robot collaboration 2015	Enable air and ground robot to collaborate on finding target of interest; reduce Marine cognitive load	Four sensors, robot operating system, developed robotic command and control, networks integration, user interface on to iPad	Four man months	$350k
Nautical chart correction process prototype 2015	Character and feature recognition, translation, and work flow support	Several databases, character recognition software, semantic interoperability, user interface	Five man months	$100k
Digital Fires 2016	Facial recognition generates call for fire to afloat platform radar and combat system	Ship combat system, missile launcher, radar, facial recognition, and ground robot operations	Three man months	$200k
Code Assessment 2018	Enable code devleopers to upload and assess code	17 different code assessment tools; semnatic interoperability between six different cyber vulnerability data sources, semantic interoperability between all of the above	Seven man months	$200k

The fourth row in Table 2 presents data for the National Geospatial Agency's NtM modernization program, which was the National Geospatial Agency's response to an increased backlog of required nautical chart updates. The EE's main objective was to increase the National Geospatial Agency's chart update speed and decrease errors in processing, which results in fewer maritime accidents. In order to accomplish this, the EE ingested existing NtMs, used translation software as necessary, checked symbol consistency, and converted the information into a stored NtM repository. The final version of the NtM program incorporated 70 languages and >100,000 symbols. Measures of performance demonstrate the application benefits of the EE in Table 3. This program demonstrated how "EE simplifies and accelerates large scale complex systems integration challenges" (Boger et al., 2016, p. 9). The EE automation tools improved the NtM update process from two weeks to thirty minutes (Boger et al., 2016, p. 9).

Table 3. EE Application Benefits in NtM Development. Source: N. Eaglestone, personal communication (2017).

Measure of Performance	Technology Integration	XML Processing(1)	PDF Optical Character Recognition(1)	Machine Language Translation(1)	Semantic Store Indexing(1)	Clean Up & Conflation Automation(1)
Objects	Text, Symbols, Imagery, and Structure	Text, Symbols, and Structure	Text, Symbols, and Structure	Text and Symbols	Text, Symbols, Imagery, and Structure	Text, Symbols, Imagery, and Structure
Productivity Gain	3x	600x	180x	240x	230x	1,000x and more
Timeliness	180 Day. ➔ 60 Day.	10 Min. ➔ < 1 Sec.	15 Min. ➔ < 5 Sec.	2 Hrs. ➔ ~30 Sec.	2.25 Hrs. ➔ ~35 Sec.	Yrs. – Hrs. ➔ Min. – Sec.
Location Accuracy	Pragmatic Representation	~99.9%(2)	~99.9%(2)	~99.9%(2)	~99.9%(2)	~100%(3)
Identification Accuracy	Pragmatic Representation	~99.9%(2)	~99.9%(2)	~99.4%(2)	~99.9%(2)	~100%(3)
Successful Detection	Pragmatic Representation	~99.9%(2)	~99.9%(2)	~99.9%(2)	~99.9%(2)	~100%(3)

The third row of Table 2 displays information pertaining to the MCWL's UTACC program, the primary goal of which was for Marines to utilize robots as teammates rather than as mere tools (Boger et al., 2016, p. 10). The EE's tools were enlisted to integrate an iPad with an unmanned aerial vehicle (UAV) and an unmanned ground vehicle (UGV) (both equipped with Light Detection and Ranging [LIDAR] sensors), and execute three primary tasks: map the area, find an object of interest, and report the object found using an advanced feature recognition tool (Boger et al., 2016, p. 10). The EE tools completed integration of these systems in three weeks, enabling the UTACC systems to accomplish all three primary tasks.

1. EE objectives and scope

The EE addresses six main areas of focus within its process, which Figure 21 depicts. These areas are formal modeling via MDD, semantic enabling, automated code generation, code quality & cyber-health, rapid integration and provisioning (via the RCPI module), and deployment. The development and implementation of the appropriate EE tools can potentially address all of the capability gaps identified in the current environment. Table 4 pairs current identified capability gaps with the appropriate EE tools.

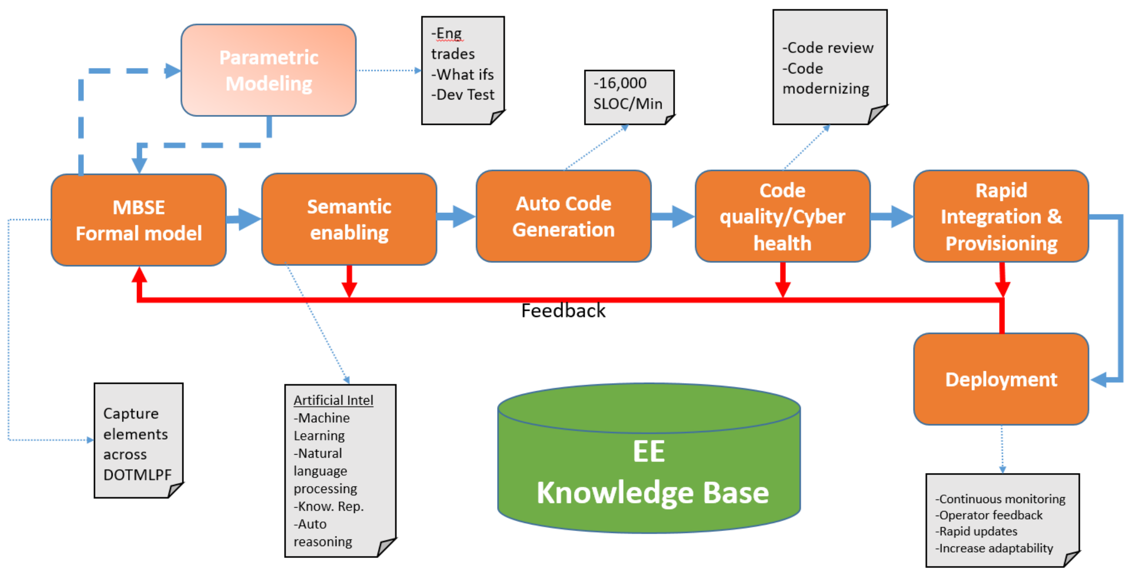


Figure 21. EE Process Overview. Source: Boger et al. (2018).

Table 4. EE Tool Implementation.

Capability Gap	EE Tool(s) used to Address Gap	Result
Outdated Document Based systems development process	MDD	System Artifacts always up-to-date; Required paperwork always up-to-date
Software Development Process	Automated Code Generation	Delivery of software code expedited
Ability to integrate systems in SoS	RCPI	Integration time and costs decreased
Lack of ability to maximize the re-use of system architectures and code due to information silos	MDD Automated Code Generation EE Knowledge Base	Ability to re-use model components in MDD from EE knowledge base; Automated Code Generation
Lack of Semantic Interoperability within SoS	Parametric modeling Semantic Enabling Rapid Integration & Provisioning	Allows for interoperability and Integration tests & use of AI, machine-learning
Lack of automated feedback and security monitoring	EE Deployment approach	Enables embedded operator feedback capture and continuous cyber monitoring in accordance with RMF instruction

2. Potential Users and Stakeholders

The EE will have the ability to support multiple users and stakeholders within the acquisitions and development communities, ranging from the program managers to the service-level development commands. Defense contractors will also be potential users and stakeholders of the EE as it becomes widely implemented. The overall end-state of the EE is to support warfighters by enabling them to rapidly develop necessary equipment that is fully interoperable and integrated within systems of systems. Again, it is crucial to emphasize that the EE can be used for almost any system development project, regardless of maturity, as long as it contains software.

3. Operational Policies and Constraints

The EE process will incorporate necessary tools to support the JCIDS, DAS, DoDAF, and RMF policies in place. The EE will support JCIDS and DAS requirements, including modular open source approach development requirements directed by the DoD Instruction 5000.2, which states:

> Program management is responsible for evaluating and implementing a modular open source approach (MOSA) to the maximum extent feasible and cost effective. This approach integrates technical requirements with contracting mechanisms and legal considerations to support a more rapid evolution of capabilities and technologies throughout the product life cycle through the use of architecture modularity, open systems standards, and appropriate business practices (Department of Defense, 2017, p. 82).

Section F of this chapter describes in greater detail further EE integration into the JCIDS and DAS processes within this CONOPS in the implementation scenarios. The EE built-in capabilities, such as MOF-to-text transforms, will generate required DoDAF paperwork and views. The EE code quality and cyber health tools will incorporate all necessary RMF requirements.

4. Mission Support Environment and DOTMLPF Requirements

The EE will require continued support from service-level development commands, program managers, and developers. Generation of the EE tools will require a significant monetary investment. Implementing EE will require institutional support to overcome organizational inertia and resistance to change. The EE, if fully implemented, will require changes across each component of DOTMLPF. Table 5 outlines some DOTMLPF requirement examples, along with the possible requirement questions the change answers.

Table 5. DOTMLPF Requirements.

Doctrine	JCIDS/DAS policy changes; RMF policy changes; (How does EE change policies?)
Organization	DoD/Service-level modeling organizations (What organization is leading EE?)
Training	Modeling training requirements (How do we produce enough modelers?)
Material	EE data storage requirements (Where is the data for EE being stored)
Leadership	Modeling organization leadership (Who is in charge?)
Personnel	Modeling professionals (Who is modeling?)
Facilities	Modeling facilities (Where will modeling be done?)

D. SUMMARY OF IMPACTS

The EE will impart both operational and organizational impacts during development and while in use. All significant innovations and organizational changes incur growing pains, but the net positive results will provide capabilities that address the identified software development shortcomings.

The EE causes multiple operational impacts that support the warfighter in the operational environment. First, the EE enables rapid capability delivery once the deployment process currently in use embraces the changes necessary to leverage it. Operators can expect not only faster delivery of software updates, but also an easier process to provide application feedback. The EE's rapid development and integration capabilities will allow warfighters to leverage technology changes faster, allowing them to become adaptable to changing environments, as recommended in General Stanley McChrystal's recent book (2015), *Team of Teams*. The RCPI tools within the EE process will increase interoperability and create a higher level of integration, since new systems can easily enter into existing ones. The EE's semantic enabling capability will provide a significant new capability. Previously, syntactically connected systems were able to pass common message types and file types, but interpreting and understanding the meaning of the various files was up to the user. Now, semantically integrated system of systems can share meaning across applications. This means far less interpretation for the operator, who can now focus on analytical thinking and making more informed decisions. Finally, the EE will positively impact cyber heath as RMF compliance, continuous threat monitoring, and security controls can be implemented quickly and efficiently. While most users will not appreciate this new-found security, the DoD operational network defenders will be most thankful. In summary, it is most likely that the operational users will experience tremendous positive impacts from implementing an EE-based approach.

Organizationally, implementing the EE will have long-term positive effects in every possible way. All performers related to development and rapid delivery will feel more connected to the operators, like they are part of the fighting team. They will feel as if their contributions, whatever their role, are critical and useful.

However, moving from the current as-is software development processes in DoD to one that leverages the EE will require a change in practices, which is the definition of innovation. Creating innovation requires that management and leadership change, and that, so far, has been difficult to foster in the DoD. Fortunately, the services are starting to recognize the severity of the inefficiencies within their current software development processes and are starting to think about model-based systems engineering and software development operations. These are both good and necessary, but not sufficient to achieve the EE vision. The good news is that their recognition of a problem is the first step in creating change in leadership, creating a sense of urgency.

E. IDEAS FOR NON-MATERIEL CHANGES

In developing a concept of operations for a new capability such as the EE, it is first important to analyze if any changes to DOTLPF might instead be able to cause that change, since these changes tend to be less expensive than material solutions. The purpose of DOTLPF alternatives is to determine if non-material solutions effectively address the capabilities gaps. Below in Table 6 is analysis that supports the development of EE based on a lack of viable DOTLPF alternatives.

Table 6. DOTLPF Changes Analysis

Doctrine	Changes to doctrine will not eliminate or significantly reduce capability gaps.
Organizational	Organizational changes cannot eliminate capability gaps. Changes to organizational structures could eliminate some information silos.
Training	Possible changes to training could demonstrate positive impacts in the current development and acquisitions environment but will not provide the necessary improvements to eliminate current capabilities gaps.
Leadership & Education	Education of leaders and identifying current capabilities gaps can assist in optimizing current development and acquisitions efficiencies but cannot fully realize the needed transformation
Personnel	Changes to force structure or hiring additional personnel cannot eliminate current capabilities gaps.
Facilities	Additional or improved facilities cannot eliminate current capabilities gaps.

F. IMPLEMENTATION SCENARIO MAPS

This section of the CONOPS proposes implementation scenario maps that present development and documents requirements during the DAS and JCIDS process timelines and highlight areas the EE will have positive impact. This scenario assumes that EE tools have been fully developed and the necessary support environment is mature.

1. Pre-Systems Acquisition Period

The first implementation scenario map, which Figure 22 displays, presents the pre-systems acquisition period, which typically ends after fulfillment of Milestone B exit criteria. The pre-systems period consists of two DAS phases: Material Solution Analysis (MSA) Phase and Technology Maturation and Risk Reduction (TMRR) Phase.

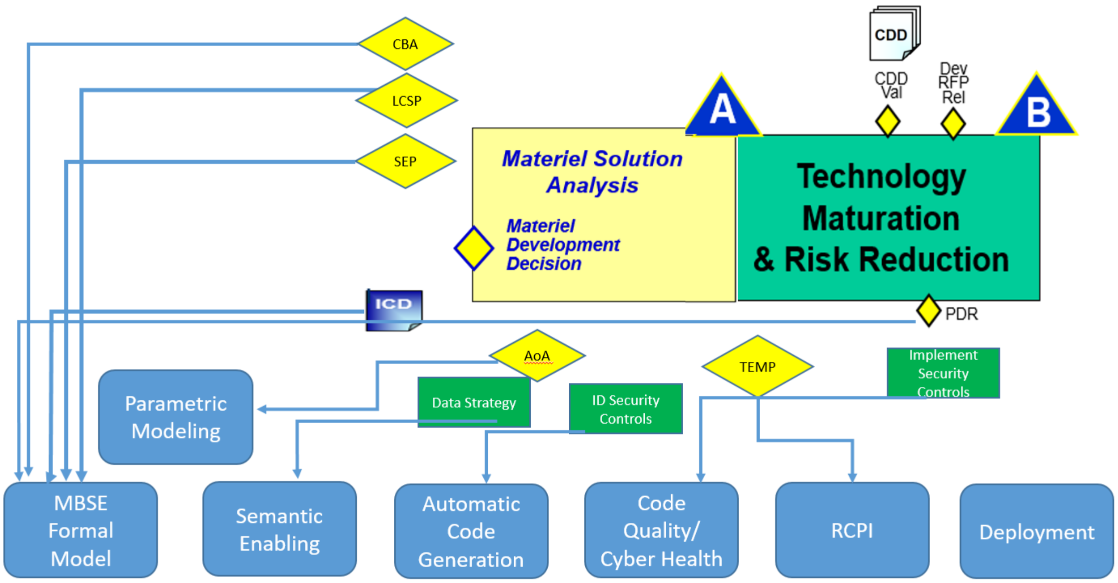


Figure 22. Pre-Systems Acquisition Period Implementation Scenario Map. Adapted from Fast, class lecture notes (2017).

The EE's MBSE Formal Model tools play an instrumental role in the improvement of many development requirements during this period, including:

- Capabilities-Based Assessment (CBA)
- Life Cycle Sustainment Plan (LCSP)
- Systems Engineering Plan (SEP)
- Initial Capabilities Document (ICD)
- Preliminary Design Review (PDR)

Parametric modeling tools assist in the execution of AoA, and data strategy designs exploit semantic-enabling tools. Automatic code generation tools identify necessary security controls. The RCPI platform tools integrate into the test and evaluation master plans (TEMP) and security controls are implemented using the EE code quality and cyber health tools.

2. Program Initiation Period

The Program Initiation Period follows the Pre-Systems Acquisition Period, after the program successfully completes Milestone B exit criteria. This period encompasses the DAS Engineering & Manufacturing Development (EMD) phase. This phase focuses on the development, building, and testing of the product to ensure all requirements are met to transition to the actual production and deployment phase. Figure 23 maps EMD phase requirements to recommended EE process tools. The MBSE Formal Model tools assist in the development requirements of the Test Readiness Review (TRR), Critical Design Review (CDR), and Post CDR assessments. The TEMPs are updated with parametric modeling tools, providing critical what-if analysis. The RCPI platform tools demonstrate SoS interoperability during Post PDR assessments.

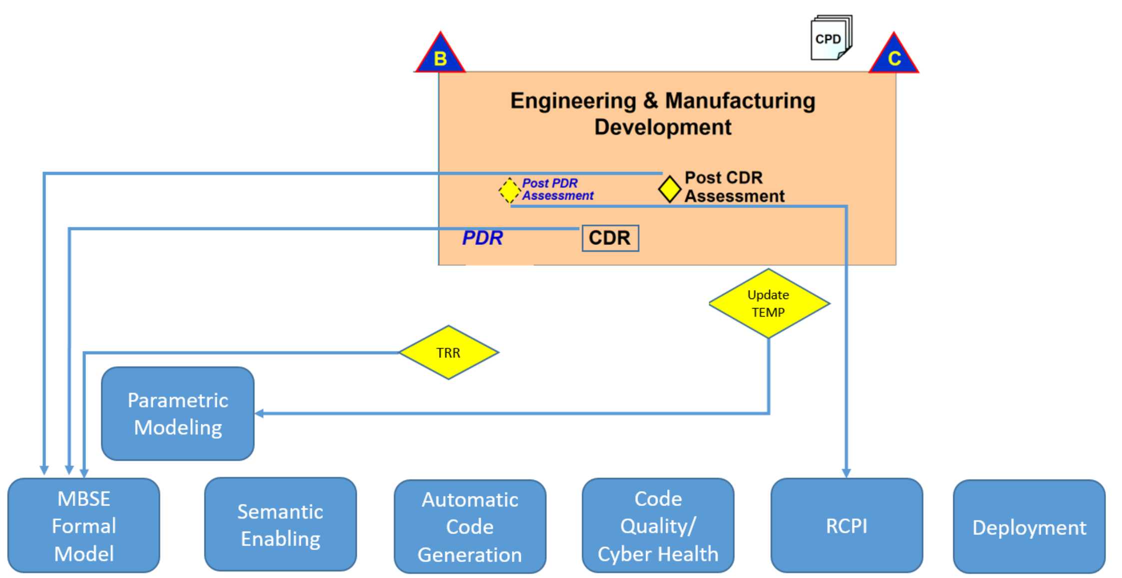


Figure 23. Program Initiation Period Implementation Scenario Map. Adapted from Fast, class lecture notes (2017).

3. Production Decisions Period

The Production Decisions Period is the final period of the implementation scenario. Figure 24 depicts this final period in the developmental process. During this period, the DAS Production & Deployment (P&D) Phase and Operations & Support (O&S) Phase occur. Low-rate initial production (LRIP) begins in this phase, with RCPI tools continuing to improve interoperability as the new systems are introduced into the SoS. Parametric modeling tools provide requisite simulation tools to support initial operational test and evaluation (IOT&E) requirements. Finally, as P&D transitions to full O&S, MBSE Formal Models continue to support life-cycle sustainment.

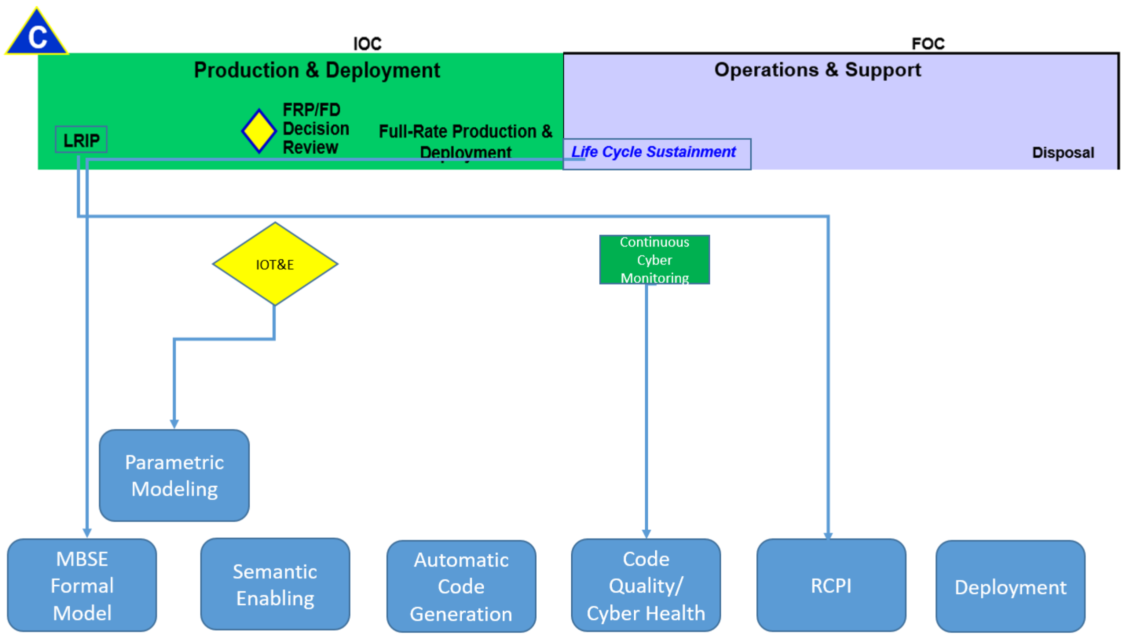


Figure 24. Production Decisions Period Implementation Scenario Map. Adapted from Fast, class lecture notes (2017).

G. ANALYSIS OF PROPOSED SYSTEM

The final section of this CONOPS is an analysis of the EE and the potential capability gaps it could address. This CONOPS has identified many potential improvements; however, there are also disadvantages and limitations associated with the EE. Determining whether to adopt any new system requires an examination of potential alternatives and possible trade-offs prior to moving past the Pre-Systems Acquisition Period.

1. Summary of Improvements

This CONOPS has identified many improvements the EE could provide. Six proofs of concept have demonstrated that the existing EE tools work. Moreover, the EE tools are able to solve six capability gaps. The implementation scenario maps identify many positive impacts, satisfying nineteen JCIDS and DAS developmental requirements.

2. Disadvantages and Limitations

Implementing the EE may fundamentally change the organizational make-up of the entire JCIDS, DAS, and sustainment infrastructure. Such change is hard. There will be people who want to "own" the EE and make it a proprietary money-maker; the DoD must insist that the EE remains a government-owned set of tools and standards.

Another "disadvantage" is that initially the EE process does not produce faster results than any other process, though there is much higher certainty of success (which is crucial, since an estimated 50–60% of all software development projects fail). The big payoff is that any upgrades and new, incremental integration requirements can be achieved much faster than before, making continuous improvement and security possible.

3. Alternatives and Possible Trade-offs

The USAF is already implementing something called the Pivotal Cloud Foundry (PCF) to execute development operations (DEVOPS), which sounds very similar to the EE. Research of this effort shows that this DEVOPS approach is similar to the RCPI platform, yet lacks the formal models that enables the requirements generation, lacks sophisticated information security techniques, and makes no claim to conduct any integration, even at the technical or syntactic levels. Consider the EE as DEVOPS on steroids, with much more capability. Still, the resistance to even using the PCF in the Air Force is a good lesson learned for adoption of the EE in the DoD. If the program that the USAF used the PCF on had not been a failure for at least the last decade, it is quite possible that the systems program office (SPO) would not have tried PCF.

The Navy is starting to embrace MBSE, at least in some developmental levels. Systems engineers are using SYSML in development, but they are far from approaching the capabilities inherent to those in the EE.

THIS PAGE INTENTIONALLY LEFT BLANK

V. CONCLUSION, RECOMMENDATIONS, AND AREAS FOR FURTHER RESEARCH

This thesis proposed a CONOPS using the EE in the development, integration, and sustainment life-cycles of future systems. Although some of the described EE tools are currently only conceptual in nature, six proofs of concepts have confirmed all the existing tools and capabilities within the EE. It is reasonable to assume that the conceptual tools will be similarly effective when built.

A. SUMMARY OF RESULTS

The research and development of the CONOPS for the EE identified several principal takeaways. The first is the recognition that all types of engineering disciplines are quickly adopting MBSE, MDD, and ontologies. This shift toward the use of formal models and intricate ontologies is a direct result of quickly emerging technologies and standards that address increased complexity and the proliferation of SoS. Preventing cognitive overload and capturing the enormous amount of intricate details associated with the development of these complex SoS requires formal models. Formal models work—they reduce costs, accelerate delivery, and improve operational performance.

The second takeaway is that the EE would enable the military to become an operationally adaptive force. The EE accomplishes this by providing forces the capability to adjust material solutions, such as SoS integration, in close to near-real time. The EE increases adaptability by enabling rapid requirements collection, which transitions to prompt capability delivery. During this process, the EE leverages legacy applications and data sources for rapid integration as well as platform provisioning that facilitates the reuse of previously developed models. This speed to capability reduces costs and improves overall performance.

Finally, EE improves the security of code in the increasingly challenging cyber environment. This is especially true as new capabilities and systems enter into the SoS that may present security vulnerabilities. The EE would provide the ability to apply rapid risk management to address and incorporate necessary corrections into the system almost simultaneously.

B. REMAINING CHALLENGES

There are challenges that remain in order to successfully implement the EE into DoD systems and software development processes. The first remaining challenge is completing the development of the EE tools. Overcoming this challenge requires necessary funding and support, which leads to a follow-on challenge: convincing an organization within the DoD to embrace this new method in order to incorporate the powerful tools to modernize its development processes. Overcoming organizational inertia and reluctance to change is difficult, but it is possible with the right leaders who believe the risk is worth the rewards.

The final significant challenge is training and educating the modelers, systems engineers, and decision makers. Developing formal models is an arduous task requiring highly trained individuals. These individuals are in high demand and are expensive to contract. In order to successfully implement the EE, a large contingent of modelers organic to the DoD who can build formal models with the necessary depth and complexity is paramount. As the CONOPS identified, contracting these modeling teams initially would be DoD's only option, but future research should examine an internal, long-term solution. Moreover, because the current set of systems engineers do not often conduct formal modeling, improving their mastery of these methods will require some work.

C. FURTHER RECOMMENDATIONS

Development of the EE's CONOPS resulted in several recommendations for ensuring successful EE implementation. They include:

- Leaders within the DoD responsible for capabilities development need to participate in the OMG. The primary reason for this is the fact that leaders within industry are constantly collaborating to develop standards and concepts within the OMG. Many of the tenets and concepts of the EE developed as a result of OMG participation. Participation in the OMG ultimately allows for DoD to provide feedback and help guide future standards that will be beneficial to future endeavors;

- The DoD should support the further development of the EE's tools in order to evolve its capabilities, allowing it to reach maximum potential. Finalizing the EE's tools will allow the DoD to use the EE with increasingly complex systems.

D. FUTURE RESEARCH

This thesis has identified that future research on MDD is necessary. While there has been significant information and research on MBSE, there are research gaps regarding MDD. Future research could possibly entail detailed case-study generation in order to discover recently developed implementations of MDD processes and the associated lessons learned.

The study of the future use of ontologies and their benefits in MBSE is another possible area for future research. As this thesis proposed, the EE intends to take advantage of ontologies built using AML. There is much that can be learned from researching and capturing the processes of developing ontologies and demonstrating how to leverage the information within them. This becomes more important as many new algorithms based on artificial intelligence methods emerge.

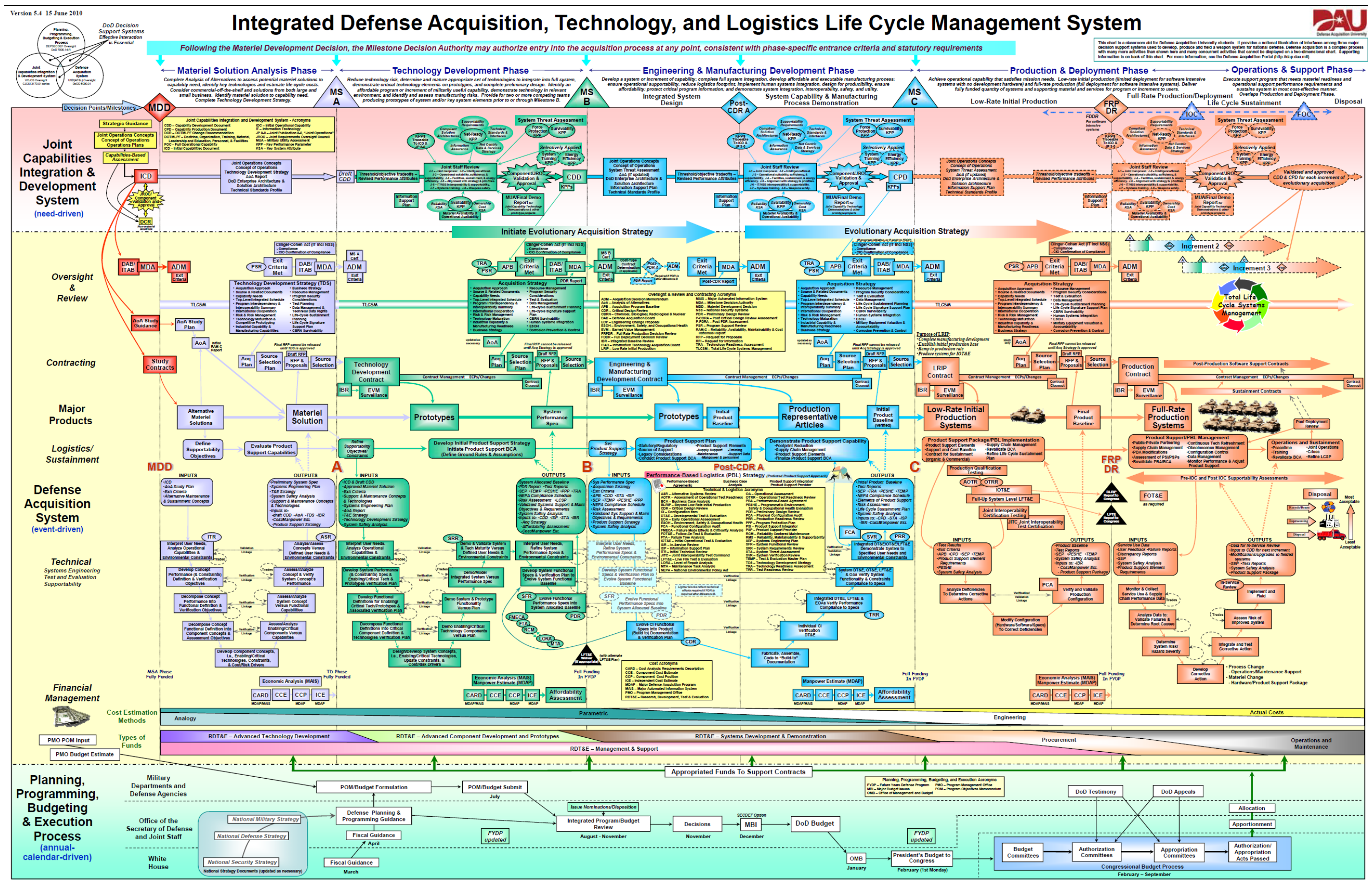
Manned-unmanned teaming is a new field within robotics in which developers attempt to create interdependence between teams of humans and unmanned systems. There is an engineering extension called interdependence analysis that generates requirements for such manned-unmanned teaming. Further research could determine how interdependence analysis might evolve as a process within the EE.

The final area of future research should conduct a detailed comparison of the costs and benefits associated with the acquisition of the necessary modelers for the EE. This analysis should include estimation of costs associated with training, educating, and retaining expert modelers resident within DoD compared to the alternative course of action of contracting the services of outside modelers. Possible analysis could include current as-is DoD modelers, fiscal break-even point cost comparisons, and other non-fiscally related advantage-disadvantage comparisons.

E. CONCLUSION

Engineering design and development processes have evolved from hand-drawn sketches to complex, digital models across all fields of engineering. After researching this evolution, the DoD should embrace this transition and further invest in model-based development in its pursuit of future complex SoS. This thesis provided the background, concepts, and analysis of incorporating the EE in future software development and presented an executable CONOPS to integrate the EE within the current acquisitions environment and practices.

APPENDIX. INTEGRATED DEFENSE ACQUISITION, TECHNOLOGY, AND LOGISTICS LIFE CYCLE MANAGEMENT SYSTEM



Source: Defense Acquisition University (2010)

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Defense Acquisition University. (2010, June 15). Integrated defense acquisition, technology, and logistics life cycle management system wall chart version 5.4. Retrieved from https://timemilitary.files.wordpress.com/2013/02/acquisition_life_cycle_management.pdf

Department of Defense. (2015, December 18). *Manual for the operations of the Joint Capabilities Integration and Development System (JCIDS)* (CJCSI 3170.011). Washington, DC: Author. Retrieved from http://www.jcs.mil/Portals/36/Documents/Library/Instructions/3170_01a.pdf?ver=2016-02-05-175022-720

Department of Defense. (2017, August 10). *Operation of the Defense Acquisition System* (DoD Instruction 5000.2). Washington, DC: Author. Retrieved from http://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/500002_dodi_2015.pdf

Department of Defense. (n.d.). Background – DODAF – DoD architecture framework version 2.02 – DoD Deputy Chief Information Officer. Retrieved August 29, 2018, from https://dodcio.defense.gov/Library/DoD-Architecture-Framework/dodaf20_background/

Boger, D., Eaglestone, N., Miller, S., Deleot, C. & S. Rosa. (2018). Enabling operationally adaptive forces. Paper presented at the 2018 Acquisitions Research Symposium, Monterey, CA. Retrieved from https://www.researchsymposium.com/conf/app/researchsymposium/unsecured/file/343/SYM-AM-18-066-013_Boger.pdf

Boger, D., Eaglestone, N., Miller, S., & Deleot, C. (2016, December). *Transforming integration & interoperability: How to radically reduce complexity in the evolution of our mission critical systems*. Unpublished.

Edson, R., & Frittman, J. (2010, May 12). *Illustrating the CONOPs continuum and its relationship to the acquisition life cycle*. Presented at Acquisition Research Symposium, Monterey, CA. Retrieved from https://calhoun.nps.edu/bitstream/handle/10945/33465/NPS-AM-10-032.pdf?sequence=1

Fosse, E. (2016, January). *Model-based systems engineering MBSE 101 JPL*. Presented at the INCOSE IW. Retrieved from http://www.omgwiki.org/MBSE/lib/exe/fetch.php?media=mbse:2016_iw-mbse_101.pdf

Fosse, E., Devereaux, A., Harmon, C., & Lefland, M. (2015). Inheriting curiosity: Leveraging MBSE to build Mars2020. *American Institute of Aeronautics and Astronautics*. Retrieved from https://trs.jpl.nasa.gov/bitstream/handle/2014/45871/15-3486_A1b.pdf?sequence=1

Friedenthal, S., Moore, A., & Steiner, R. (2008). *A practical guide to SysML: The systems modeling language*. Burlington, Mass: Elsevier/Morgan Kaufmann.

Institute of Electrical and Electronics Engineers. (1998). *IEEE Guide for information technology - System definition - Concept of operations (ConOps) Document* (IEEE Standard 1362–1998). Retrieved from https://doi.org/10.1109/IEEESTD.1998.89424

International Council on Systems Engineering (INCOSE). (2007). *INCOSE systems engineering vision 2020* (No. INCOSE-TP-2004-004-02) (p. 32). International Council on Systems Engineering (INCOSE). Retrieved from http://www.ccose.org/media/upload/SEVision2020_20071003_v2_03.pdf

Mukerji, J., Miller, J. (2003). *MDA guide version 1.0.1*. Retrieved from https://www.omg.org/cgi-bin/doc?omg/03-06-01.pdf

Kim, P. K. (2018). *Accelerating the risk management framework* (Master's thesis). Retrieved from Naval Postgraduate School Restricted Collection.

Malone, R., Friedland, B., Herrold, J., & Fogarty, D. (2016). Insights from large scale model based systems engineering at Boeing. *INCOSE International Symposium*, *26*(1), 542–555. Retrieved from https://doi.org/10.1002/j.2334-5837.2016.00177.x

Mittal, S., Technologies, D., Zeigler, B. P., & Risco-Martin, J. L. (2009). Implementation of a formal standard for interoperability in M&S/systems of systems integration with DEVS/SOA, *The International C2 Journal, 3*(1), 1–63.

NDIA Systems Engineering Division, M&S Committee. (2011). *Final Report of the Model Based Engineering (MBE) subcommittee*. Washington, DC: NDIA. Retrieved from https://www.ndia.org/-/media/sites/ndia/meetings-and-events/3187-sullivan/divisions/systems-engineering/modeling-and-simulation/reports/model-based-engineering.ashx.

Object Management Group. (n.d.-a). About OMG | Object Management Group. Retrieved August 15, 2018, from https://www.omg.org/about/index.htm

Object Management Group. (n.d.-b). MDA specifications | Object Management Group. Retrieved August 23, 2018, from https://www.omg.org/mda/specs.htm

Object Management Group. (n.d.-c). OMG standards are the future of military technology | Object Management Group. Retrieved August 15, 2018, from https://www.omg.org/industries/military.htm

Object Management Group. (2015). *OMG MDA guide version 2.0*. Retrieved from https://www.omg.org/cgi-bin/doc?ormsc/14-06-01.pdf

Rainey, L. B., & Tolk, A. (2015). *Modeling and simulation support for system of systems engineering applications*. Hoboken, New Jersey: John Wiley & Sons, Inc.

Tolk, A., Diallo, S. Y., & Turnitsa, C. D. (2007). Applying the levels of conceptual interoperability model in support of integratability, interoperability, and composability for system-of-systems engineering. *Systeics, Cybernetics, and Informatics*, *5*(5), 65–74.

Tolk, D. A., & Muguira, J. A. (2003). The levels of conceptual interoperability model. In *2003 Fall Simulation Interoperability Workshop, number 03F-SIW-007* (pp. 14–19).

Walden, D. D., Roedler, G. J., Forsberg, K., Hamelin, R. D., Shortell, T. M., & International Council on Systems Engineering (Eds.). (2015). *Systems engineering handbook: A guide for system life cycle processes and activities* (4th ed.). Hoboken, New Jersey: Wiley.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California