



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**USING SYSTEM ARCHITECTURE, REVIEW ENTRY
CRITERIA, AND STANDARD WORK PACKAGE DATA
TO ENABLE RAPID DEVELOPMENT OF INTEGRATED
MASTER SCHEDULES**

by

Burton W. Porter Jr.

March 2016

Thesis Advisor:
Second Reader:

Rama D. Gehris
Ronald R. Carlson

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE March 2016	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE USING SYSTEM ARCHITECTURE, REVIEW ENTRY CRITERIA, AND STANDARD WORK PACKAGE DATA TO ENABLE RAPID DEVELOPMENT OF INTEGRATED MASTER SCHEDULES			5. FUNDING NUMBERS	
6. AUTHOR(S) Burton W. Porter Jr.				
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. IRB Protocol number N/A .				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) While engineers must participate in the construction of the Integrated Master Schedule, this thesis proposes a way to reduce that effort through automation. When standardized sub processes exist, automated task name construction with consistent action/object naming convention can be applied to multiple system artifacts. These repeating sub processes also allow the derivation of task sequence and dependencies. The Architecture-Based Utility for Repeating Task Planning (A-BURTP) is a Microsoft Access database constructed of tables containing system configuration items, artifact data, local processes, and governing instructions listed in the systems engineering plan (SEP). A-BURTP generates task planning worksheets (TPWs) in Microsoft Excel that import directly into Microsoft Project. The TPWs contain all task attributes needed to allow schedule developers to work independently to link, sort, and group the imported tasks. Appendices include research on system context for Naval Air Systems Command (NAVAIR) systems onboard aircraft carriers and research on Model-Based Systems Engineering data exports using Vitech CORE9. Scope is limited to constructing the artifact tasking for an Engineering Change Proposal on a fictional system under change. NAVAIR processes and instructions are used when available. The 2011 Office of the Secretary of Defense SEP template is a key reference.				
14. SUBJECT TERMS ims, mbse, swp, sep, scheduling, IGS, automation, standard work, NAVAIR, cost, ECP, engineering change, estimating, system architecture, task planning, project, critical path, EVM, project management, systems engineering, CVN, obsolescence, integrated master schedule, portfolio schedule			15. NUMBER OF PAGES 137	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**USING SYSTEM ARCHITECTURE, REVIEW ENTRY CRITERIA, AND
STANDARD WORK PACKAGE DATA TO ENABLE RAPID DEVELOPMENT
OF INTEGRATED MASTER SCHEDULES**

Burton W. Porter Jr.
Civilian, Department of the Navy
B.S., Embry Riddle Aeronautical University Worldwide, 2011

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SYSTEMS

from the

**NAVAL POSTGRADUATE SCHOOL
March 2016**

Approved by: Rama D. Gehris, Sc.D.
Thesis Advisor

Ronald R. Carlson
Second Reader

Ronald E. Giachetti, Ph.D.
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

While engineers must participate in the construction of the Integrated Master Schedule, this thesis proposes a way to reduce that effort through automation. When standardized sub processes exist, automated task name construction with consistent action/object naming convention can be applied to multiple system artifacts. These repeating sub processes also allow the derivation of task sequence and dependencies.

The Architecture-Based Utility for Repeating Task Planning (A-BURTP) is a Microsoft Access database constructed of tables containing system configuration items, artifact data, local processes, and governing instructions listed in the systems engineering plan (SEP). A-BURTP generates task planning worksheets (TPWs) in Microsoft Excel that import directly into Microsoft Project. The TPWs contain all task attributes needed to allow schedule developers to work independently to link, sort, and group the imported tasks.

Appendices include research on system context for Naval Air Systems Command (NAVAIR) systems onboard aircraft carriers and research on Model-Based Systems Engineering data exports using Vitech CORE9. Scope is limited to constructing the artifact tasking for an Engineering Change Proposal on a fictional system under change. NAVAIR processes and instructions are used when available. The 2011 Office of the Secretary of Defense SEP template is a key reference.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	BACKGROUND	1
1.	Integrated Program Schedules.....	1
2.	Schedule Development	1
3.	Engineering Change Proposal Cost and Schedule Estimates.....	3
4.	System Artifact Tasks	4
5.	Methods of Building Schedules	5
B.	PROBLEM STATEMENT.....	10
C.	RESEARCH QUESTIONS	10
1.	Primary Research Question.....	10
2.	Supporting Research Questions	11
D.	SCOPE, LIMITATIONS, AND ASSUMPTIONS	11
1.	Thesis Scope.....	11
2.	Limitations.....	12
a.	<i>Savings not Quantified</i>	12
b.	<i>Initial Schedule Construction Only</i>	12
c.	<i>Simulated Data</i>	12
3.	Assumptions	12
II.	LITERATURE REVIEW	15
A.	AIRCRAFT CARRIER INFORMATION TECHNOLOGY CONTEXT	15
1.	Commander, Naval Air Forces.....	15
2.	Increasing Aircraft Carrier Forward Presence: Changing the Length of the Maintenance Cycle	17
3.	Carrier Planning Activity (CPA) Carrier Availability Schedule.....	18
B.	HIGHER-LEVEL GOVERNMENT/NAVY GUIDANCE AND INSTRUCTION.....	20
1.	Work Breakdown Structures for Defense Material Items	20
2.	NAVAIRINST 4355.19D Systems Engineering Technical Review (SETR) Process	21
3.	OSD Systems Engineering Plan (SEP) Outline	23
C.	BEST MANAGEMENT PRACTICES	25
1.	The Agility Advantage	25
2.	Planning and Scheduling Excellence Guide	26

D.	LOCAL PROCESSES AND INSTRUCTIONS	27
1.	AIR 4.2.3 Integrated Government Schedule Development and Maintenance Toolkit.....	27
2.	NAVAIR Standard Work Packages	31
III.	METHODOLOGY.....	33
A.	CREATING AN ARCHITECTURE-BASED UTILITY FOR REPEATING TASK PLANNING	33
B.	OBJECTIVES.....	35
1.	Minimize Direct Engineering Effort to Develop IMS Tasking	35
2.	Use Standard Output Format.....	35
3.	Create Task Names.....	36
4.	Estimate Task Durations	36
5.	Assign Task Dependency	36
6.	Populate WBS Field with System Architecture Data	36
7.	Assign Engineering Review Type	36
8.	Populate CDRL Field with Artifact Data	37
9.	Populate Resource Names.....	37
10.	Populate Basis of Estimate for Original Duration.....	37
11.	Populate Team or IPT Field.....	37
12.	Populate Schedule Risk and Opportunity Field	37
C.	FUNCTIONAL DECOMPOSITION OF TASK CREATION	37
1.	“Calculate SEP Dates” Function	38
a.	<i>Create Tasks Function (Node 1.1)</i>	38
b.	<i>Link Tasks Function (Node 1.2)</i>	39
c.	<i>Calculate Task Dates Function (Node 1.3)</i>	40
2.	Decomposition of Create Tasks Function, Automated.....	40
a.	<i>Create Artifact Tasks Function</i>	43
b.	<i>Create SETR Tasks Function</i>	46
c.	<i>Create ECP Admin Event Tasks Function</i>	49
3.	Summary of Functional Decomposition	52
D.	DATA SOURCES AND STORAGE	53
1.	Introduction to Data Sources and Storage	53
2.	Background.....	53
3.	Data Storage in A-BURTP Tables	53
a.	<i>Input Elements Table</i>	53
b.	<i>Artifacts Table</i>	54
c.	<i>Processes Table</i>	55

	d.	<i>Process Steps Table</i>	56
	e.	<i>Input Select Reviews Table</i>	57
	f.	<i>Artifact Maturity Review Table</i>	58
E.		ANALYSIS OF SUCCESS IN MEETING OBJECTIVES.....	60
	1.	Minimize Direct Engineering Effort to Develop IMS Tasking	61
	2.	Use Standard Output Format.....	62
	3.	Create Task Names.....	63
	4.	Estimate Task Durations	64
	5.	Assign Task Dependency	65
	6.	Populate WBS Field with System Architecture Data	67
	7.	Assign Engineering Review Type	68
	8.	Populate CDRL Field with Artifact Data.....	68
	9.	Populate Resource Names.....	68
	10.	Populate Basis of Estimate for Original Duration.....	68
	11.	Populate Team or IPT Field.....	69
	12.	Populate Schedule Risk and Opportunity Field	69
F.		METHODOLOGY SUMMARY	69
IV.		CONCLUSIONS.....	73
A.		RESULTS AND RECOMMENDATIONS.....	73
	1.	Results.....	73
	a.	<i>Uninterrupted Engineering Assessments</i>	73
	b.	<i>Concise and Unambiguous Task Names</i>	75
	c.	<i>Schedule and Systems Engineering Plan Agreement</i>	75
	d.	<i>One-time Entry for Repeating Process Steps</i>	75
	2.	Recommendations.....	75
	a.	<i>Use a Database Tool to Create Schedule Tasks</i>	76
	b.	<i>Use a Database for SEP Sections 4.3, 4.4, 4.5, 4.6</i>	76
	c.	<i>Review and Update Local Processes</i>	77
	d.	<i>Export from Model-Based Systems Engineering Tools When Available</i>	77
B.		DISCUSSION OF RESEARCH QUESTIONS	78
	1.	Primary Research Question.....	78
	2.	Supporting Research Questions	78
C.		BENEFITS OF RESEARCH.....	80
	1.	Reduce Non-Value-Added Activity.....	80

2.	Rapid Response and Enhanced Agility	80
3.	Integration of Management Tools	81
D.	TOPICS FOR FUTURE RESEARCH	81
1.	Improvements to A-BURTP	81
a.	<i>Include Test Events</i>	81
b.	<i>Evaluate Complexity and Risk</i>	81
c.	<i>Improve the Tool Design</i>	83
2.	Review NAVAIRINST 4355.19e and Explore the Ability to Incorporate the Release Backlog Review (RBR)	84
3.	Smash Bureaucracy	84
4.	Cost Estimation	86
APPENDIX A.	MODEL-BASED SYSTEMS ENGINEERING EXPORTS USED IN RAPID SCHEDULE DEVELOPMENT	89
A.	MODEL-BASED SYSTEMS ENGINEERING (MBSE)	89
B.	CREATING A SAMPLE SYSTEM ARCHITECTURE	92
C.	LITERATURE FOR APPENDIX A	97
1.	DOD Architectural Framework 2.02	97
2.	NASA WBS Handbook	97
3.	Vitech CORE9 Architecture Definition Guide	98
4.	Vitech CORE9 System Definition Guide	98
5.	Using Innoslate for Program Management	99
D.	MBSE TO IMS TO SEP DATA REUSE	99
E.	APPENDIX A SUMMARY	100
APPENDIX B.	NAVAIR CVN SOS CONTEXT	103
A.	NAVAIR ELECTRONIC SYSTEMS ON CVNS	103
B.	CVN AVAILABILITY FOR MAINTENANCE	104
C.	COTS OBSOLESCENCE	105
D.	PROGRAM CONTEXT—IRRESISTIBLE FORCES AND IMMOVABLE OBJECTS	106
E.	PMA-251 CONFIGURATION MANAGEMENT PLAN (CMP) FOR LAUNCH AND RECOVERY EQUIPMENT	108
F.	OBSOLESCENCE-DRIVEN DESIGN REFRESH PLANNING FOR SUSTAINMENT-DOMINATED SYSTEMS	109
LIST OF REFERENCES		111
INITIAL DISTRIBUTION LIST		115

LIST OF FIGURES

Figure 1.	The A-BURTP Method of Task Creation Compared with Current Methods of Collecting Task Data and Calculating Dates for the Systems Engineering Plan (SEP dates).....	7
Figure 2.	U.S. Navy Pacific and Atlantic Carrier Fleets (Source: CNAF 2014)	16
Figure 3.	Sample Section of Past CPA CVN Schedule (Adapted from CPA 2013).....	19
Figure 4.	SEP Outline Format (Adapted from OSD 2011a, 1)	23
Figure 5.	Task Planning Worksheets Method	29
Figure 6.	“Engineering Interview” Method.....	30
Figure 7.	Methodology for A-BURTP	35
Figure 8.	Non-Automated IDEF0 “Calculate SEP Dates”	39
Figure 9.	Create Tasks Function Using A-BURTP.....	41
Figure 10.	Create Artifact Tasks Function	44
Figure 11.	Create SETR Tasks Function.....	48
Figure 12.	Create ECP Admin Event Tasks Function.....	51
Figure 13.	IMS Tasks Imported from A-BURTP.....	66
Figure 14.	A-BURTP Tasking Linked.....	67
Figure 15.	A-BURTP Query Producing Artifact Tasks	70
Figure 16.	Notional Timelines.....	74
Figure 17.	Schedule Risk Assessment (Adapted from OSD 2011a).....	83
Figure 18.	“Document the Model, Don’t Model the Document” (Source: Vitech 2012)	86
Figure 19.	Proposed Data Flow	87
Figure 20.	Mapped MBSE Architecture and Artifact Data.....	91
Figure 21.	System Context for “System A”	93
Figure 22.	“System A” System Architecture Hierarchy	94
Figure 23.	Network Switch.....	95
Figure 24.	Relationships between Architecture and Artifacts	96
Figure 25.	IMS and SEP Review and Changes (Source: LaGrone 2014)	105
Figure 26.	Interactions between Internal and External Elements.....	107

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

Table 1.	PDR Entry Criteria (Adapted from NAVAIR 2008).....	21
Table 2.	NAVAIRINST 4355.1D “System Requirements and Capabilities” (Adapted from NAVAIR 2008).....	22
Table 3.	“Input Elements” Table from A-BURTP	54
Table 4.	“Artifacts” Table from A-BURTP	55
Table 5.	“Processes” Table from A-BURTP.....	56
Table 6.	“Process Steps” Table from A-BURTP	57
Table 7.	“Input Select Reviews” Table from A-BURTP	58
Table 8.	“Artifact Maturity Review” from A-BURTP.....	59
Table 9.	“SETR Object Steps” Table from A-BURTP	60
Table 10.	Task Planning Worksheet Data Sources.....	61
Table 11.	Microsoft Project Standard Fields Assigned to NAVAIR 4.2.3 Data Dictionary Are Populated by A-BURTP.....	63
Table 12.	Task Name, Task Dependency, and System Architecture Data from A-BURTP	64
Table 13.	A-BURTP Duration Formulas	65
Table 14.	MBSE Architecture Elements Assigned “Documented by” Relationships.....	90

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

A-BURTP	Architecture-Based Utility for Repeating Task Planning
CDRL	contract data requirements list
CDR	critical design review
CI	configuration item
CM	configuration management
CMP	configuration management plan
CNAF	Commander Naval Air Force
COTS	commercial off-the-shelf
CPA	Carrier Planning Activity
CSG	Carrier Strike Group
CVN	carrier vessel, nuclear
CVW	Carrier Air Wing
DAU	Defense Acquisition University
DODAF	Department of Defense Architecture Framework
DPIA	docking planned incremental availability
ECP	engineering change proposal
EOA	end of availability
FF	finish-to-finish scheduling logic
FS	finish-to-start scheduling logic
GASP	Generally Accepted Scheduling Principles
IDEF	integration definition
IGS	integrated government schedule
IMS	integrated master schedule
IPT	integrated product team
MBSE	model-based systems engineering
NAVAIR	Naval Air Systems Command
NDIA	National Defense Industrial Association
NMCI	Navy Marine Corps Intranet
NVA	non value-added
OSD	Office of the Secretary of Defense

PASEG	Planning and Scheduling Excellence Guide
PDR	preliminary design review
PIA	planned incremental availability
PM	program manager
RCOH	refueling complex overhaul
SE	systems engineer
SEP	systems engineering plan
SETR	systems engineering technical review
SOA	start of availability
SOS	system of systems
SRA	schedule risk assessment
SRR	system requirements review
SVT	schedule visibility task
SWP	standard work package
TPW	task planning worksheet
TRR	test readiness review
WBS	work breakdown structure

EXECUTIVE SUMMARY

The 2012 National Defense Industrial Association (NDIA) Planning and Scheduling Excellence Guide (PASEG) discusses the integration of management tools and declares that integration must start between the systems engineering plan (SEP) and the Integrated Master Schedule (IMS). NDIA further encourages integrating the IMS with requirements management systems. This thesis explores automation between the SEP and the IMS and develops the Architecture-Based Utility for Repeating Task Planning (A-BURTP) tool.

A-BURTP is a tool constructed using Microsoft Access that stores data called for in the 2011 Office of the Secretary of Defense (OSD) SEP template and combines it with standard work process steps to derive IMS tasks. A-BURTP exports in Microsoft Excel task planning worksheet (TPW) concept (NAVAIR [4.2.3] 2010) used by Naval Air Systems Command (NAVAIR).

A-BURTP creates consistent and concise task names that follow the naming conventions called for in the PASEG. Other derived task data includes resource information, task sequence and dependency information, risk and opportunity notes, and other attributes. The A-BURTP TPW imports directly into Microsoft Project.

The concept is most useful for programs that encounter many engineering change proposals (ECPs) and require continual IMS updates. A-BURTP allows engineers to focus on their technical assessment of the ECP and then transmit the list of affected architecture configuration items to the schedule developer. A-BURTP creates tasks for the affected system artifacts associated with the configuration items. These artifacts are then associated to work steps from local procedures and the IMS tasks are created using concise action/object naming conventions.

A-BURTP allows the schedule developer to rapidly and independently create, import, link, group, and sort the new tasks. This decreases the

interruptions to engineers for cost and schedule information. Schedule reviews and tailoring can take place after the initial schedule is constructed.

References

NAVAIR [4.2.3]. 2010. Integrated Government Schedule Development and Maintenance Toolkit NAVAIR [4.2.3].

ACKNOWLEDGMENTS

The author would like to acknowledge:

Professor Ronald Carlson, Dr. Rama Gehris, and Dr. Warren Vaneman for advising and guiding this thesis.

Kenny Beverly, Christopher Cardone, Gail Edwards, Carolyn Holguin, Keith Jordon, Chris Matthews, Erik Merk, Lou Williams, Robert Rossi, Anthony Rosso, Dr. Russell Shannon, Ellen Sherman, Jonathan Simbol, Ragini Sridhar, and Kenneth White for providing insight into NAVAIR 4.8 engineering processes.

Stephanie Alcott, Christine Arsego, Lisa Heuser, Bruce Koontz, William Korosec, Eileen Lang, Michael Lawlor, Randy Lewis, Ryan Lewis, Eric Miller, Dan Nega, Lynn Parker, Robert Preedy, Jeff Scher, and Jeff Upton for their guidance in NAVAIR 4.2 cost and schedule processes.

Martha Spicer for mentoring and encouraging me to further my education.

Ernest W. Fox (CAPT, USCG, in memoriam) for his inspiration and guidance in problem solving and continual process improvement.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

1. Integrated Program Schedules

The Integrated Master Schedule (IMS) and Integrated Government Schedule (IGS) are very similar in construction and function. The IGS is used to track government actions and integrate contractor IMS(s) within the program. In-house government programs may use a very detailed IGS similar to a contractor IMS. For the purposes of this thesis, the term “IMS” was used exclusively but the principles described herein can be transferred to IGS development where detailed government tasking must be developed.

Development and maintenance of the program schedule is an administrative activity that requires a combination of scheduling best practices and engineering information. Schedule developers (schedulers) can follow best practices, but require engineering information concerning the task names, durations, resources and dependencies in order to produce or modify the schedule. Engineers can develop schedules, but their specialized degrees, expertise and certifications are needed at the same time for solving the difficult problems of assessing system architecture and designing and testing of systems and components.

2. Schedule Development

The technical approach for completion of engineering work scope is described in the systems engineering plan (SEP). The SEP is a living document that is updated with required dates for interface agreements and forecasted dates for completion of key events and (OSD 2011a). These dates come from the IMS which is a model of the technical approach (NDIA 2012). The SEP dates come from the IMS and are a result of calculating the aggregate durations of the technical approach and assessing schedule impacts to and from external touchpoints of the plan. Getting the detailed work scope plan from the engineer's

head into the scheduling tool so that all parties can understand their roles and interactions is the work of IMS development.

Engineers play a vital role in the development of the program IMS and varying degrees of engineering involvement can exist even within the same program. This can range from reuse of existing IMS components as templates that can be copied with minimal engineering input, up to situations where the engineers actually use the scheduling tool to input their own tasks.

The Naval Air Systems Command (NAVAIR) uses schedulers to construct schedules for in-house government programs. The scheduler is ultimately responsible for conformance with scheduling best practices but may not have all the information needed to create and sequence the tasks without engineering input. Meanwhile, the engineers may have the best understanding of the actual work but may not be familiar with scheduling tools or best practices (NAVAIR [4.2.3] 2010).

The determining factors in the level of engineering involvement in IMS development are often the degree to which the scheduler has access and/or knowledge of the engineering tasking and dependencies, and the degree of expertise the engineers have with the scheduling tools. Small teams may not budget for a scheduler and the engineers could have to develop the schedule themselves. Ultimately, the overall scheduling system must include adequate engineering discipline concerning the engineering processes and adequate scheduling discipline concerning the application of scheduling best practices. When the IMS is constructed with only partial engineering knowledge, work scope could be missing or true logic could be misrepresented. Likewise, if the scheduling best practices are not followed, the IMS may not function correctly with regard to date calculations once the schedule becomes active and status changes are incorporated.

3. Engineering Change Proposal Cost and Schedule Estimates

A situation where the ability to automate IMS task development becomes particularly advantageous is one where a system requires many engineering change proposals (ECPs) and many “engineering” or “bottoms-up” estimates. These programs must continually update the program IMS with detailed engineering activities since cost and schedule impacts must be evaluated prior to authorization to begin work. Cost and schedule impact assessments take time to perform and are required early in the ECP process. During these early stages of an ECP, engineers must assess the system and determine required design revisions and test events. Requests for cost and schedule information can interrupt and compete with this higher-value engineering work.

Since the system already exists and is being changed through the ECP process, many of the artifacts describing the system are already developed and the changes to these artifacts must be estimated as well. The Defense Acquisition University Acquipedia states that engineering estimates require a “substantial amount of time and effort” and require WBS data.

The engineering or “bottoms-up” method of cost analysis is the most detailed of all the techniques and the most costly to implement. It reflects a detailed build-up of labor, material and overhead costs. Estimating by engineering is typically performed after Milestone C (i.e., Low Rate Initial Production (LRIP) approval) when the design is firm, minimal design changes are expected to occur, data is available to populate the Work Breakdown Structure (WBS), drawings and specifications are complete and production operations are well-defined in terms of labor and material....

Engineering cost estimates can be quite accurate since they are usually exhaustive in covering the work to be performed by the virtue of using the work breakdown structure. These estimates also make use of insight into the specific resources and processes used in performing the work. However, a substantial amount of time and effort is required to produce and document such an estimate, making it impractical to use this method for all elements of an acquisition program's costs. (DAU 2015)

While DAU states that engineering estimates are “impractical to use for all elements of a program” (2015), some programs need to continually update engineering estimates due to continual system changes. One such example is NAVAIR information systems onboard aircraft carriers which is described in Appendix B. These embedded systems operate as parts of a system of systems (SOS) and can require many ECPs due to both changes in capability need or obsolescence of commercial off-the-shelf (COTS) hardware and software components. The timeslots available to bring the system down for upgrade are also extremely limited by the Carrier Planning Authority (CPA) carrier availability schedule. While the CPA scenario is not entirely unique, shipboard programs require significant coordination technically, chronologically, and financially to ensure interoperability of all interacting systems.

Automating portions of the estimating process can significantly reduce the time and effort. As stated by DAU, the estimates “make use of insight into the specific resources and processes” but research and documentation take time (2015). Capturing these processes uniformly in database tables allows them to be reused for future estimates. If the system component and artifact data is also stored in database tables, queries can be developed to rapidly derive IMS tasks and provide a major portion of the ECP estimate.

The one-time effort of associating system components to artifacts, artifacts to change processes, change processes to work steps, and work steps to review entry criteria should be available to the scheduler from the program SEP. If a formal SEP is not presently available, the effort of capturing this data will, at the very least, move the program towards compliance with SEP section 4 (OSD 2011a).

4. System Artifact Tasks

The Office of the Secretary of Defense (OSD) 2011 Systems Engineering Plan template states expectations for each mandated section (2011a). Technical

reviews, system artifacts, and change management are discussed in Sections 4.4 and 4.5 (2011a).

Section 4.4 requires the plan for technical reviews to include a review table (19). The review table must include entry and exit criteria and the “products/artifacts” coming from the review. Since the review must be “event-driven” (24), the projected date for each review (23) must come from an achievable plan to accomplish the work scope.

Section 4.5 of the template discusses change management control and requires a list of artifacts that describe the system baselines. Section 4.5.1 calls for a description of the change management process for the artifacts (25).

ECPs involve changes to system artifacts and the processes to create or update these artifacts contain work steps that must be added to the IMS in order to determine the schedule impact of the ECP. Engineers must evaluate the engineering change requirements against the current functional and system architectures and then determine the artifact updates required. The work steps to be completed, the sequence of the work steps, and how the artifacts relate to entry criteria for the technical reviews must all be considered in creating IMS task names, durations, resources, dependencies and other task attributes. These detailed work steps must be incorporated into the IMS in order to assess schedule impact for the ECP.

5. Methods of Building Schedules

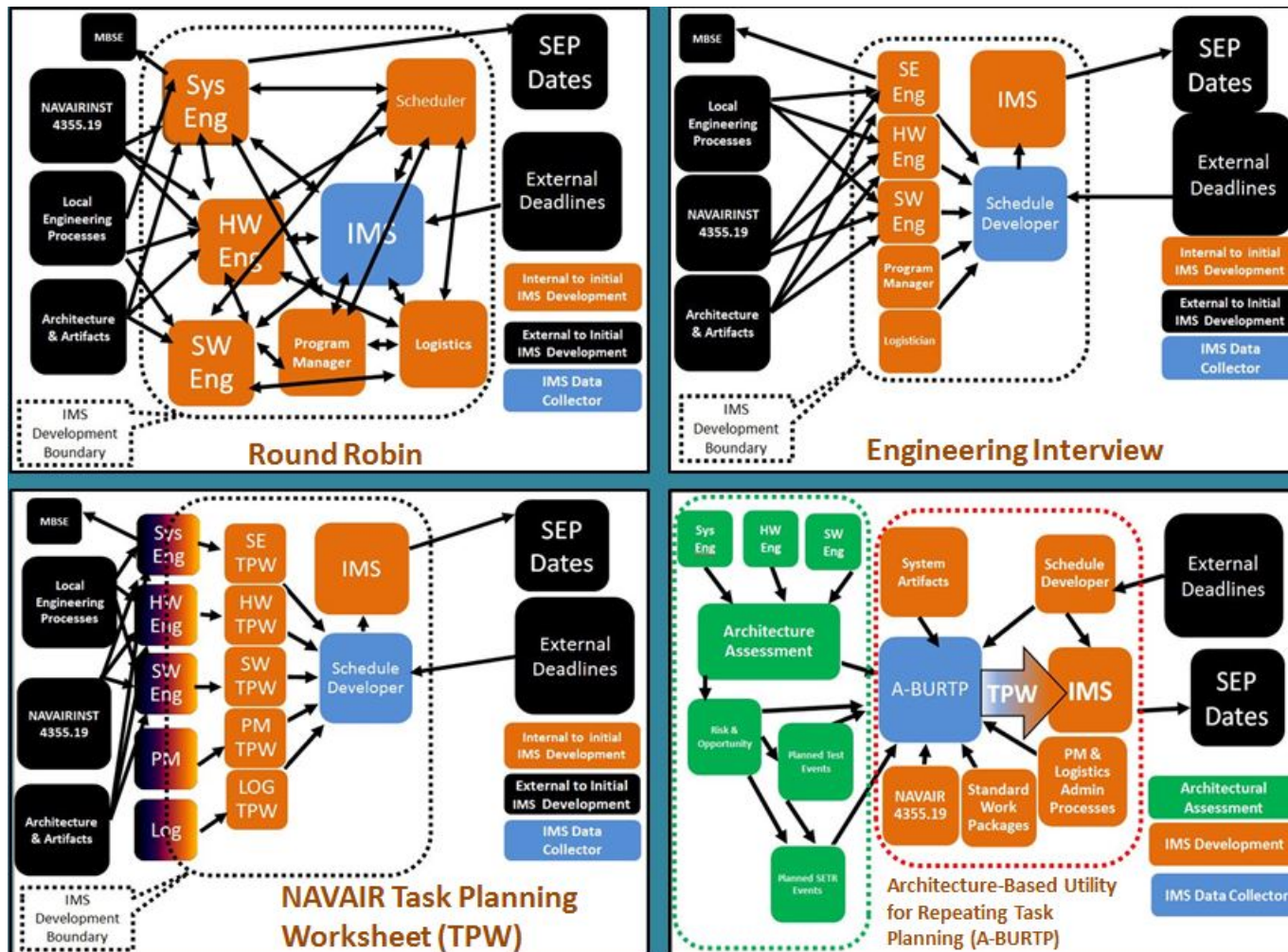
There are a variety of ways to combine the engineering and scheduling disciplines and each has merit in certain circumstances. Figure 1 illustrates three currently used methods along with the improved method developed in this thesis. Each method ultimately produces dates for the IMS.

In the “Round Robin” method (Figure 1, upper left), engineers add their own tasks to the IMS and link the tasks themselves. This direct engineering input into the IMS can also allow schedule construction errors and duplicated or

missing tasks. The Engineering Interview method (Figure 1, upper right) reduces scheduling errors, but is time consuming for both engineers and schedulers. The Task Planning Worksheets (TPW) method (Figure 1, lower left) ensures the scheduler has responsibility for task entry, but also requires a lot of detail from the engineers to populate the TPW (NAVAIR [4.2.3] 2010).

The fourth method (Figure 1, lower right) uses the A-BURTP tool developed for this thesis. A-BURTP stores all data needed for TPW population and allows the schedule developer to run queries that create the correct tasks based on the system changes required. Instead of pulling engineers into schedule development, A-BURTP creates the tasks based on the output of a preliminary engineering assessment of changes to system architecture.

Figure 1. The A-BURTP Method of Task Creation Compared with Current Methods of Collecting Task Data and Calculating Dates for the Systems Engineering Plan (SEP dates).



Instead of pulling engineers into IMS development, A-BURTP quickly derives schedule tasks from architecture assessments.

In more detail, “Round Robin” is a method used when a scheduler requires a high degree of engineering input to understand the work scope and processes. The upper left quadrant of Figure 1 illustrates the Round Robin methodology with engineers inside the boundary of schedule development. The engineers interpret guidance as it applies to a particular ECP and create IMS tasks directly in the IMS file. The IMS is routed from engineer to engineer to add in their tasks and logic links and returned to the scheduler for review, questions, and corrections.

To use this method, engineers must be fairly proficient with the scheduling software. Since Microsoft (MS) Project is common software available on the Navy Marine Corps Intranet (NMCI), many NAVAIR engineers are fairly proficient at using the product. It can be a fast way to create an initial IMS module but can also require many corrections after review by the scheduler. Beside the potential for rework and missing scope, it also places the majority of the burden for schedule development on the engineering staff. In schedules produced this way, engineers are internal to initial IMS development and the IMS itself is the central data repository. Development uses direct input from hardware, software, and system engineers as well as program managers and logisticians. Configuration management of the IMS can also become difficult with so many entities reviewing and making changes.

In one IMS created this way, review of the results showed a variety of task-naming conventions and also a variety of terms to describe the same actions or objects. Instances of duplicate tasks were not readily identified because of this inconsistent naming. The schedule structure had a high instance of relationship logic other than Finish-to-Start (FS). Start-to-Start (SS) and Finish-to-Finish (FF) logic ties were used, and both positive and negative lags were found in the logic. While all these practices are acceptable in certain instances, many of the uses had potential to allow milestone delays to go undetected once the schedule received status inputs.

These unconventional logic ties required evaluation by the scheduler and follow-up interviews with the engineers before they could be corrected. Using engineering personnel for IMS development not only reduced their availability for more critical engineering work, but also produced an IMS that needed substantial scheduling, program management, and logisticians rework including additional engineering interviews.

While the Round Robin method has some shortcomings, it avoids some of the inefficiencies incurred by the Engineering Interview method. This method is shown in the upper right quadrant of Figure 1 and uses engineers to dictate task entries to a scheduler who collects and inputs the data during team meetings. Since the engineers are still within the boundary of IMS development, they are pulled away from higher value engineering duties while attending schedule meetings. Discussions between two people while several others sit and observe can become very costly and wasteful. Even decisions concerning task names to accurately describe the actions and objects within the IMS can take significant time. This method is discussed further in the Literature Review section of this thesis.

The NAVAIR TPW method (NAVAIR [4.2.3] 2010) improves on both the Round Robin and the Engineering Interview. The TPW method helps to keep the scheduling tools in the hands of the scheduler and the engineers out of the schedule meetings. It is NAVAIR's approved method of capturing the important information that a scheduler needs to work independently to construct a schedule (2010).

The lower left quadrant of Figure 1 represents the TPW method and shows the IMS development boundary passing through the nodes that represent the engineers and other team members. This is because the TPW method requires engineers and other team members to fill out very detailed worksheets and send them to the scheduler for incorporation into the schedule. The amount of information needed to populate a NAVAIR TPW can still be a burden on the

engineer. The TPW method, although an improvement, once again pulls engineers from higher value activities to work on cost and schedule products.

The Literature Review section of this thesis includes more discussion on the engineering interview and TPW methods. These methods put the scheduler in the loop earlier and should have less rework than Round Robin, but neither is data-driven and both still require significant engineering time. A data-driven method that allows the scheduler to work independently with minimal disruption to the engineers would reduce the amount of time an engineer would need to devote to this effort.

Since many engineering efforts follow standard processes for developing, maturing, and approving artifact documents, the actions of the action/object task name could be derived from these processes. Also, since most system components and interfaces have governing artifacts that follow these process steps, the objects in the action/object task name could be derived from system architecture. A one-time research effort to map system architecture elements to artifacts and then map artifacts to processes could be used to automate TPW population. This is the concept shown for the A-BURTP method in the lower right quadrant of Figure 1.

B. PROBLEM STATEMENT

A method is needed to allow schedule developers to determine artifact tasks and task sequence with minimal disruption to higher-value engineering efforts.

C. RESEARCH QUESTIONS

1. Primary Research Question

Can IMS tasking be derived from sources other than direct engineering input?

2. Supporting Research Questions

What data is available and useful to schedule developers?

What other data is needed to construct IMS tasks?

Can IMS task creation be automated?

Can predecessor/successor information be determined?

Can task names be standardized?

Can task durations be estimated?

What other IMS task attributes can be derived from available data?

D. SCOPE, LIMITATIONS, AND ASSUMPTIONS

1. Thesis Scope

The scope of this thesis is to determine and demonstrate if a method to reduce the involvement of engineers in the creation of an IMS can be created. Scope is focused on the demonstration of the creation of bottom-level tasking for a single ECP within the IMS of a single program. The scope is further narrowed to only those tasks associated with preparation for technical reviews. While the concept is applicable to other types of tasking, the scope of this thesis is narrowed to demonstrate the concept, and future research is encouraged to experiment with capability such as in tasks involving test events, procurements, and logistics products. The main focus herein was on demonstrating a process that automatically derives the artifact tasks and their relation to technical review entry criteria.

This thesis utilizes three key documents that already exist at NAVAIR: the Systems Engineering Technical Review (SETR) process to determine review entry criteria (NAVAIR 2008), the NAVAIR Integrated Government Schedule Development and Maintenance Toolkit which provides TPW guidance (NAVAIR ([4.2.3] 2010), and NAVAIR Standard Work Packages (SWP) which detail work required to develop engineering products (NAVAIR [4.0] 2015). Similar guidance

from other organizations could be substituted as long as continuity of business rules is maintained.

2. Limitations

a. *Savings not Quantified*

No studies on time to develop IMS tasks were performed in this thesis or any of the research literature. The fact that adequate schedule development information is able to be collected semi automatically without disrupting engineers is assumed to be better than current methods employed. Future work to quantify potential time/effort savings is left for another researcher.

b. *Initial Schedule Construction Only*

Only a limited set of business rules are explored in this thesis. Removing engineers from the tedious work of initial task creation does not release them from all schedule development involvement. Review by engineers of the new IMS tasking would still be required prior to running Schedule Risk Assessments (SRA).

c. *Simulated Data*

This study is limited to a small portion of a fictional system. This fictional system is similar to an aircraft carrier SOS. SWP steps are simulated and actual organizational directives would need to be evaluated and estimated. Initial population of process and process step tables would require audit of actual documentation within an organization.

3. Assumptions

This thesis assumes a system exists and is undergoing change. While the methodology is applicable to an emerging system, some level of system definition needs to be completed before applicable data can be identified. This development and/or analysis is performed by engineers. The intended use of A-BURTP is to allow engineers to perform an uninterrupted initial assessment of

the system architecture and provide required data to schedulers who, in turn, can use the tool to determine the artifact tasks required for SETR entry.

THIS PAGE INTENTIONALLY LEFT BLANK

II. LITERATURE REVIEW

A. AIRCRAFT CARRIER INFORMATION TECHNOLOGY CONTEXT

1. Commander, Naval Air Forces

Review of the Commander, Naval Air Force (CNAF) webpages helps in understanding the CVN IT program context boundary as it relates to the forward presence and tactical air power CVN capabilities. CVNs are the center of each Carrier Strike Group (CSG) which also include a Carrier Air Wing (CVW), and a complement of surface and submarine ships (CNAF n.d.).

Forward presence without tactical air power does not meet the capability need, nor does tactical air power without forward presence. And, while it requires a CSG to provide both capabilities in a given radius, a single CSG still does not fully meet the capability need (Yardley et al. 2008). Therefore, it is the CNAF complement of Pacific and Atlantic CSGs that provide the capabilities. The 11 CVNs shown in Figure 2 are used by CNAF to combine with CVWs and escorts to create specific capabilities for specific periods to meet the needs.

Figure 2. U.S. Navy Pacific and Atlantic Carrier Fleets
(Source: CNAF 2014)



The information on the CNAF public webpage reveals high-level architecture and required capability. It reveals how the CVN IT program context boundaries are related to the boundaries of the CVN complement. Since the IT system is embedded in the CVN and is needed for the CVN to meet the capability needs, any interfaces between the CVN IT system and other CSG elements must be considered. These interfaces can include a variety of CVW configurations as well as other CVN IT systems. It naturally follows that changes to interfacing systems, CVW complement, or CVN cyber security requirements can create ECPs for the CVN IT system, and all schedules must be coordinated to construct the complete CSG.

2. Increasing Aircraft Carrier Forward Presence: Changing the Length of the Maintenance Cycle

Increasing Aircraft Carrier Forward Presence: Changing the Length of the Maintenance Cycle describes the planning of the CVN maintenance availabilities (Yardley et al. 2008). Yardley et al. discuss the complexity of CVN subsystems and the need to make the CVN available for subsystem maintenance. The global capability of forward presence must continue even while a few of the CVNs are down for maintenance.

Fleet schedulers must balance the maintenance, training, deployment, and readiness sustainment of carriers to meet presence demands. They must also consider the overall goal of a “6+1 fleet” that has at least six carriers deployed (or able to deploy) within 30 days, and a seventh carrier deployed (or able to deploy) within 90 days. (Yardley et al. 2008)

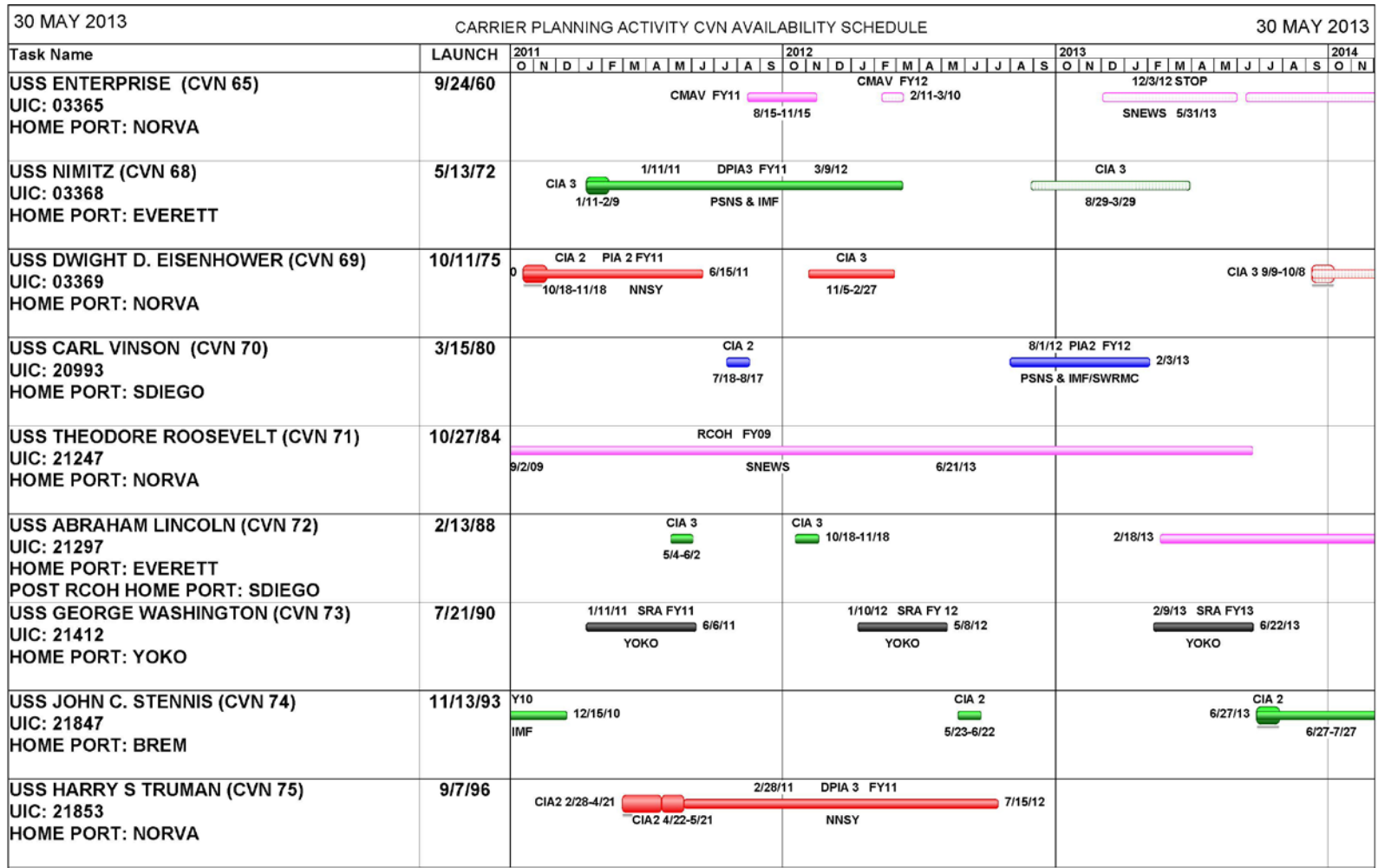
Each CVN's 50-year service life is divided roughly in the middle by a Refueling and Complex Overhaul (RCOH) event that takes around three years to perform. This is a major opportunity for subsystem programs to perform maintenance as well. While 25 years is an adequate service interval for nuclear refueling, other systems onboard require much more frequent servicing and upgrades. Shorter opportunities exist during Planned Incremental Availability (PIA) and Docking Planned Incremental Availability (DPIA) events. The CPA schedulers manage the CVN schedule and provide Start of Availability (SOA) and End of Availability (EOA) dates which govern maintenance opportunities for embedded systems. (Yardley et al. 2008)

Because meeting the capability requirements depend on “6+1 fleet” operational availability, subsystems that limit operational availability must follow the same goal. Therefore, CVN subsystems inherit the schedule constraints of the CPA availability schedule along with the requirements that flow down from the CNAF capabilities of Forward Presence and Tactical Air Power.

3. Carrier Planning Activity (CPA) Carrier Availability Schedule

Figure 3 contains a sample of a past CPA CVN availability schedule showing the time periods when maintenance and upgrades can be performed. There is a start of availability (SOA) and end of availability (EOA) for each bar illustrated on the schedule (CPA 2013).

Figure 3. Sample Section of Past CPA CVN Schedule (Adapted from CPA 2013)



Sample schedule shows maintenance availabilities with start and end dates.

The CPA CVN availability schedule uses colored bars to illustrate the time periods when maintenance can be performed. The white space between the bars is the deployable period where the CVN is able to provide capability (Yardley et al. 2008). System architectures for CVN IT programs must interoperate with the forecasted CSG architecture of the same deployable periods. CVN IT architecture must be sustainable for the forecasted deployment periods (Singh and Sanborn 2008), and development must be started early enough to allow installation and test prior to EOA. The CVN CPA schedule frames the schedule context for all CVN subsystems and creates deadlines for the IMS.

B. HIGHER-LEVEL GOVERNMENT/NAVY GUIDANCE AND INSTRUCTION

1. Work Breakdown Structures for Defense Material Items

Section 1.5.3 a. and b. of MIL-STD-881C *Work Breakdown Structures for Defense Material Items* define WBS as follows:

- a. A product - oriented family tree composed of hardware, software, services, data, and facilities. The family tree results from systems engineering efforts during the acquisition of a defense materiel item.
- b. A WBS displays and defines the product, or products, to be developed and/or produced. It relates the elements of work to be accomplished to each other and to the end product. In other words, the WBS is an organized method to breakdown a product into sub - products at lower levels of detail. (DOD 2010, 4)

MIL-STD-881C specifies a product oriented WBS and states, “The Program WBS and Contract WBS aid in documenting the work effort necessary to produce and maintain architectural products in a system life cycle” (DOD 2010, 1). Further, the standard describes how the WBS is used as a “coordinating medium” for cost, schedule, technical, and performance data (DOD 2010, 2). This is not to say that the WBS defines the order of completion, but that when all lower levels of a WBS are complete, the next higher level is complete.

MIL-STD-881C establishes the relationship between architecture products and work product definition. Architecture data could be of value in IMS task creation if it can be associated correctly with action-oriented work steps.

2. NAVAIRINST 4355.19D Systems Engineering Technical Review (SETR) Process

For Naval Aviation programs, NAVAIR instruction (NAVAIRINST) 4355.19d Systems Engineering Technical Review (SETR) Process (NAVAIR 2008) is the authority on SETR events. It describes the purpose, expectations, and timing of each review and lists associated entry and exit criteria for each as well. The instruction calls for SETRs to be event driven rather than schedule driven and provides clear business rules which can be tailored by systems engineers for specific program situations (NAVAIR 2008).

The explicit SETR entry criteria for each review can be captured in table format, stored in a database, and queried to provide task attributes indicating predecessor and successor information to the schedule developer. This is demonstrated in A-BURTP. Table 1 and Table 2 are examples of entry criteria for Preliminary Design Review (PDR). Table 1 contains the major heading for PDR entry criteria (NAVAIR 2008). This same data can be extracted for each review.

Table 1. PDR Entry Criteria (Adapted from NAVAIR 2008)

PDR Entry Criteria	
a.	System Requirements and Capabilities
b.	Test, Evaluation, and Certification of Product
c.	Engineering Processes, Control, and Analysis
d.	Programmatic Processes, Control, and Analysis
e.	Program Execution Risk and Performance Risk”

Major headings

Table 2 contains the PDR entry criteria listed in NAVAIRINST 4355.19D under the “a. System Requirements and Capabilities” major heading (NAVAIR 2008). As this information is available for each SETR, the information could be normalized and a table constructed using the decompositions called out in the NAVAIRINST.

Table 2. NAVAIRINST 4355.1D “System Requirements and Capabilities” (Adapted from NAVAIR 2008)

a. System Requirements and Capabilities	
(1)	Sub-system Design Specifications complete
(2)	CTEs have achieved TRL 6
(3)	IDDs matured through detailed design Interface Control Documents
(4)	Interface Control Documents between sub-systems complete
(5)	Traceability from CDD to function baseline to sub-systems specification complete
(6)	Human Systems Design Standards flowed to sub-systems
(7)	R&M diagnostics have been completely addressed in design allocations
(8)	Top Level Software Design Description and/or Software Architecture Description complete
(9)	Software IDD or equivalent complete

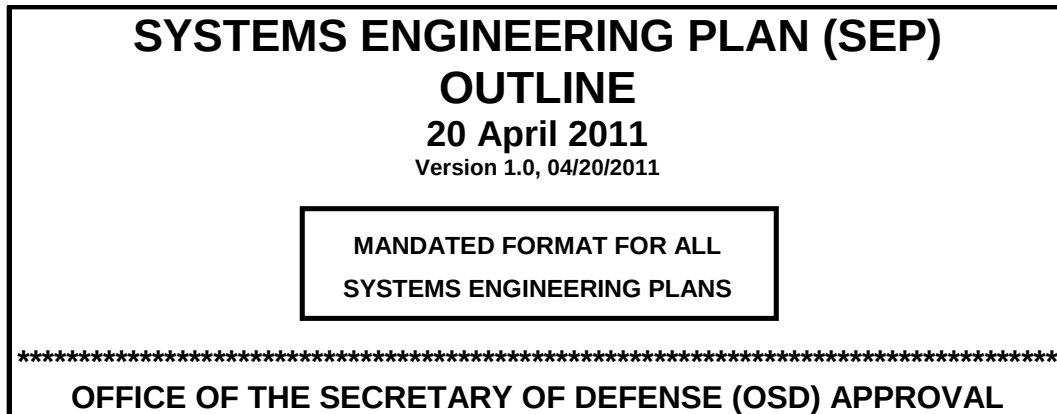
PDR entry criteria placed in table format.

The SETR process complies with the OSD SEP expectation that a “standard process [be used for] conducting technical reviews” (OSD 2011, 23). NAVAIRINST 4355.19d has a description of each review including purpose of the review and entry criteria (NAVAIR 2008). In practice, NAVAIR programs must decompose the types of artifacts called out in Table 2 into the actual system artifacts that must be developed or updated.

3. OSD Systems Engineering Plan (SEP) Outline

Figure 4 displays the cover page proclaiming the mandated format of the SEP format approved by the Office of the Secretary of Defense (OSD) on April 20, 2011. This template has placeholders for significant information and delineates expectations for population of the placeholders (OSD 2011a).

Figure 4. SEP Outline Format (Adapted from OSD 2011a, 1)



Mandated and approved by the Office of the Secretary of Defense.

Section 1 of the SEP template requires a delineation of alignment with other engineering plans, an update plan, and an update record. Section 1 also requires statements concerning update and approval authorities.

The expectations for Section 1 start with:

SEP should be a “living” “go to” technical planning document and the blueprint for the conduct, management, and control of the technical aspects of the government’s program from concept to disposal. SE planning should be kept current throughout the acquisition lifecycle. (OSD 2011a, 6)

The SEP is critical to the IMS and the IMS is critical to the SEP. The following SEP sections discuss the need for dates in the SEP (which come from the IMS) and the requirement for the SEP to define system requirements, artifacts, and change processes (which are used to create the IMS tasks).

Section 2 of the SEP provides information to, and requires information from, the IMS. Section 2.1 provides a list of architecture products that will be developed including DODAF architecture efforts, and system architecture. Methods of documenting physical and functional interfaces must be indicated as well (OSD 2011a, 7). SEP Table 2.1-1 requires dates by which each interface memorandum of agreement (MOA) signature and SEP Table 2.2-1 requires dates for certifications (OSD 2011a, 7). Schedule development requires an understanding of the interfaces in order to determine the dates by which they must be agreed to.

Section 3 concerns technical schedule and also requires input from the IMS. Section 3.1 requires key milestone dates including the SETR event forecast dates and a Schedule Risk Assessment (SRA) (OSD 2011a, 8). Section 3.2 concerns resource and cost/schedule reporting (OSD 2011a). A-BURTP produces preliminary three-point duration estimates for SRA and demonstrates the ability to populate engineering resource information.

Section 4 concerns “Technical Activities and Products” and is broken down into several sub headings that are relevant to this thesis (OSD 2011a). Section 4.3.1 requires discussion on traceability of documentation down to the configuration item (CI) level and a statement of tools used for traceability (OSD 2011a). The processes to develop mature documents, along with the required maturity levels for SETR entry, are key data elements needed for IMS development. Engineering effort to develop this table (potential from an MBSE tool) could be exported and reused for IMS construction.

Section 4.4 concerns technical reviews and states the expectation “Programs should use a standard process for conducting technical reviews” (OSD 2011a, 23). Relevance to this thesis is that the SETR entry criteria from NAVAIRINST 4355.19D are required to be displayed in a table format. Converting SETR entry criteria to table format is performed in this thesis within the tool developed.

Section 4.5 requires definition of artifacts that govern technical baselines. The section allows that a list of artifacts might have already been created in Section 4.4, but the specific requirement to create a list is carried here (OSD 2011a, 24). For the purposes of this thesis, these artifacts are objects that require actions and therefore require IMS tasks to be created.

Section 4.5 also requires definition of change processes including roles and responsibilities, change classifications, and authorities for approval (OSD 2011a, 25). These are the actions needed to construct IMS tasks for ECPs and routing processes determine predecessor and successor logic and readiness for SETR entry.

While not all sections of the SEP template were addressed as relevant to this thesis, the IMS requires much input from the SEP in order to output dates back to the SEP. There is an iterative process necessary between the IMS and the SEP. With due diligence in SEP section 4, the output can be reused and repurposed during IMS task development.

In summary, SEP section 4 contains no dates but provides a detailed plan to be followed. Aggregating the SEP section 4 tables, along with the contents of some documents they refer to—such as SETR entry criteria and the work steps in SWPs—reveals the bottom-level tasking for the IMS. The IMS becomes a model of what is projected to happen if the SEP is followed. The dates in SEP sections 2 and 3 (SEP Table 2.1-1 required MOA dates, SEP Table 2.2 expected certification dates, SEP Section 3.1 planned milestones, etc.) are calculated by the IMS.

C. BEST MANAGEMENT PRACTICES

1. The Agility Advantage

The Agility Advantage (Alberts 2011) discusses the subject of change and the ability of entities to successfully cope with change. In a chapter entitled *Defining Agility* Alberts states, “Agility is the ability to successfully effect, cope with, and/or exploit changes in circumstances” (190).

Alberts further makes the point that success does not always require agility as he writes “In situations that are stable or change in ways that do not have a significant impact on an entity’s state, entities still need to succeed, but they do not need agility to be successful” (2011, 191). So, while Alberts considers agility to be an advantage, he does not always consider it to be a necessity.

Another point worth noting is Alberts’ discussion on improving agility where he makes the point that being agile does not require perfection:

Agility can be improved by putting in place its enablers and by removing or reducing the effects of its inhibitors. Does agility require that we must be equally good at everything and under all circumstances? No. Agility does not require an entity to be equally good in a changed circumstance rather that an entity’s performance, effectiveness, and efficiency need to be satisfactory. (2011, 199)

With regard to the CVN IT system context of unpredictable externally driven changes established in this thesis, it follows that program agility would be a desirable quality for a program team. The program manager and system engineer need to be able to quickly dispatch and inform the team, assess the impact of changes, and develop strategies for successful and timely delivery of systems to CVNs.

The advantage of agility certainly applies in the macro sense on CVN IT systems. The ability to rapidly construct architectures for specific purposes and time periods would allow programs to support CNAF capability needs (Yardley et al. 2008). Likewise, the IMS maintenance process needs to be agile enough to absorb ECPs, quickly adjust strategy with minimal team disruption, and report impacts as necessary to external stakeholders.

2. Planning and Scheduling Excellence Guide

The main source of guidance on schedule development for this work comes from the June 6, 2012, release of the National Defense Industrial Association (NDIA) Planning and Scheduling Excellence Guide (PASEG).

The PASEG (NDIA 2012) begins by introducing the Generally Accepted Scheduling Principles (GASP) and goes on to provide practical methods of meeting them. This work used the GASP as a starting list of requirements for IMS development.

Section 3.3 “Integration of Management Tools” in the PASEG discusses the SEMP. The PASEG describes the importance of consistency across technical plans, the authority of the SEMP, and the IMS being a “reflection” of the SEP (24).

The PASEG warns against creating an Integrated Master Plan (IMP) until the requirements are fairly well defined and the technical approach is determined (2012).

The IMS and the SEP must contain common data points and must progress somewhat together in order to maintain this alignment. This is done through the Work Breakdown Structure (WBS) which is a product-based hierarchy coming directly from system architecture (2012).

The PASEG provides a source of traceability for IMS creation requirements. The Generally Accepted Scheduling Principles (GASP) were used to evaluate objectives in the Methodology section of this thesis.

D. LOCAL PROCESSES AND INSTRUCTIONS

1. AIR 4.2.3 Integrated Government Schedule Development and Maintenance Toolkit

While the PASEG gives high-level guidance on IMS development (NDIA 2012), the AIR 4.2.3 Integrated Government Schedule Development and Maintenance Toolkit (IGS toolkit) describes specific NAVAIR practices (NAVAIR [4.2.3], 2010). The IGS toolkit assumes a new program IGS is being developed and shows how to employ various spreadsheets to capture IGS structure data and task requirements.

The term “IGS” is used merely to provide distinction between a contractor product “IMS” and a government product “IGS.” This thesis uses “IMS” exclusively because it is a well-known term and reduces confusion.

Section 3 of the IGS toolkit discusses schedule framework with 3.1 describing the relationships between the Work Breakdown Structure (WBS), the Organizational Breakdown Structure (OBS), and the Responsibility Assignment Matrix (RAM). The toolkit states in Section 3.1: “It is the program manager’s responsibility to develop the Work Breakdown Structure” and provide it to the schedule developer (NAVAIR [4.2.3] 2010). The IGS toolkit references the MIL-HDBK-881A for WBS development. This Military Handbook has since become a Military Standard and updated to MIL-STD-881C.

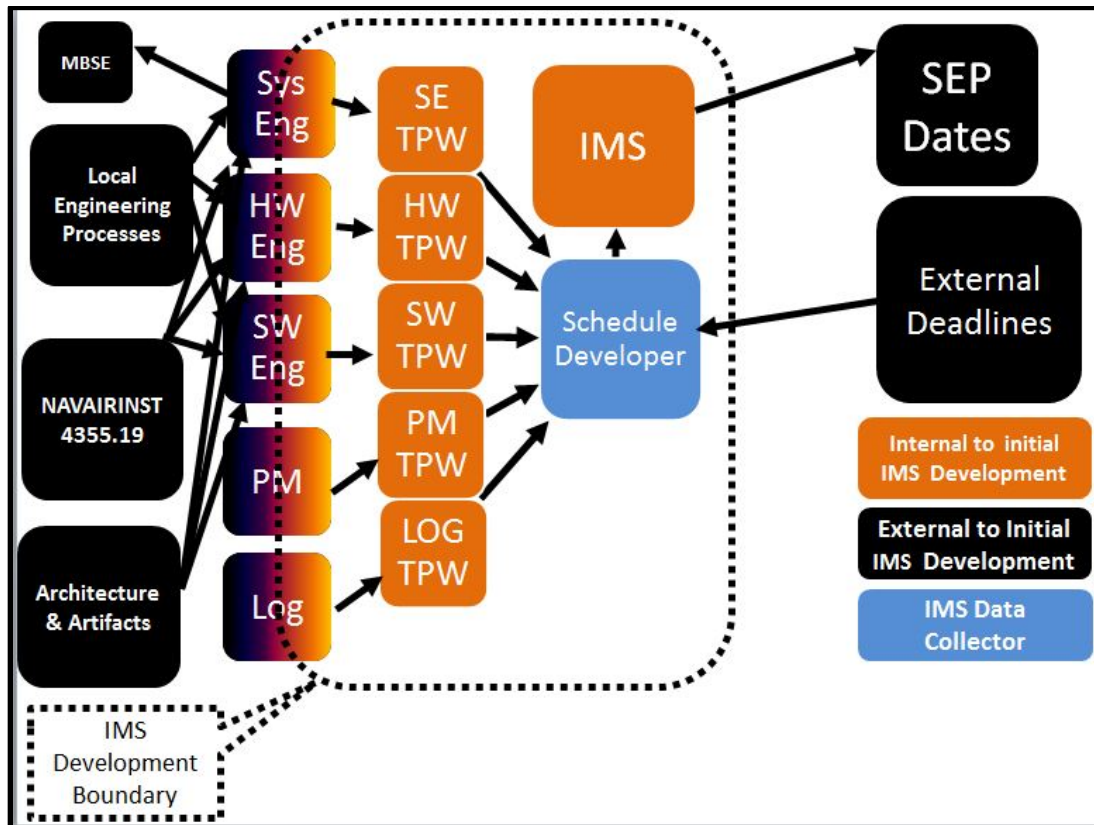
Section 3.3 addresses task definition and introduces the Task Planning Worksheet (TPW). Responsible managers on the integrated product team (IPT) receive 2–3 hours of TPW training after which they populate TPWs with discrete tasking for their area of responsibility. TPWs can be filled out independently by the manager or task planning meetings can be conducted with help from the scheduler. TPWs are MS Excel files containing rationale on duration assumptions, discussion of schedule risk, dependencies on other tasks, three-point durations for SRA, and other information. The TPW information is used by the 4.2.3 scheduler to build the IMS (NAVAIR [4.2.3] 2010).

NAVAIR 4.2.3 Integrated Program Management (IPM) includes a TPW section in their Schedule Development Tool Kit (NAVAIR [4.2.3] 2010). The benefits include the ability for multiple stakeholders to provide input at the same time and for the scheduler to maintain IMS configuration management. The TPW method moves the quality inspection forward in the process as TPW can be cycled back to the originator several times for any follow-up questions. TPWs do not require the engineers to be skilled in MS Project or scheduling best practices.

Figure 5 shows the TPW method with engineers straddling the boundary of IMS development. In this method, engineers determine which engineering

processes are applicable to the particular ECP and populate a TPW to send to the schedule developer.

Figure 5. Task Planning Worksheets Method

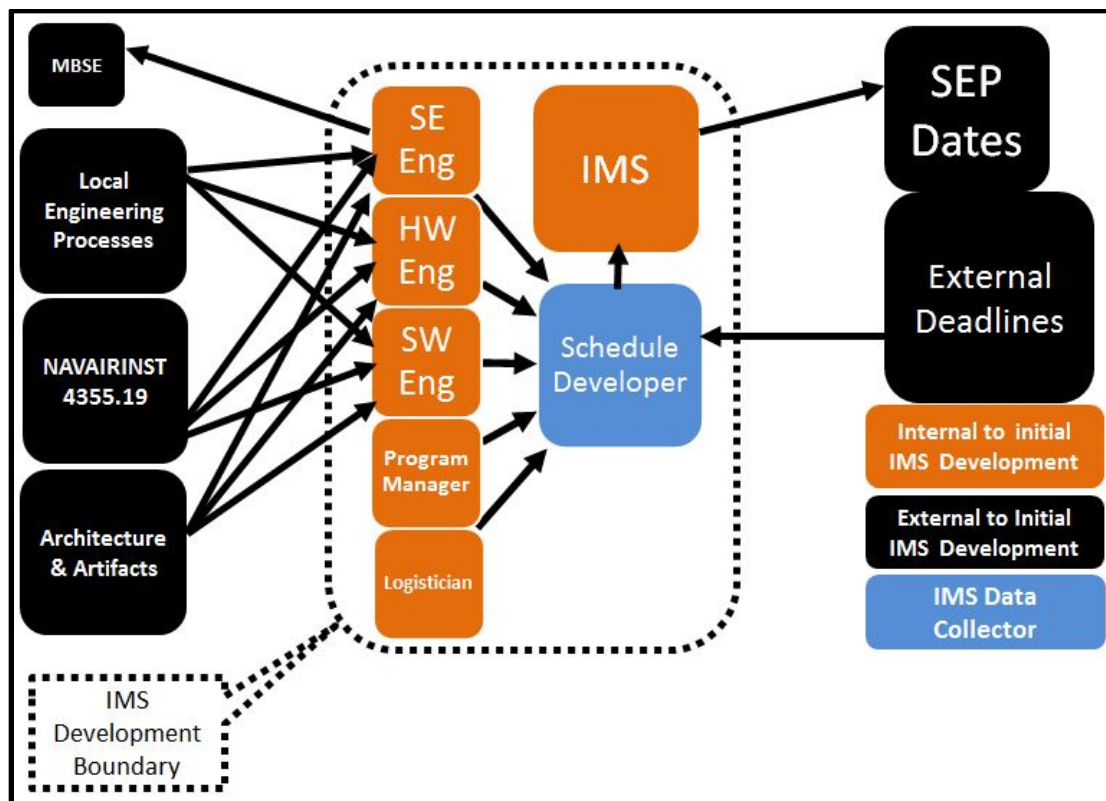


Engineers are partially within IMS development since they must provide detailed information on the TPW.

The use of TPWs requires most of the work of IMS task creation to be performed by engineers while linking the tasks is performed by the scheduler. However, TPWs require the dependency information for each task to enable the schedule developer to properly link them in the correct sequence within the IMS module. Determining and entering this dependency information is also performed by the engineers and other stakeholders. It is a cleaner process than emailing a schedule around to the engineers but still uses engineering expertise to fill out the TPWs instead of performing value-added design work.

Figure 6 illustrates the Engineering Interview method of schedule development. This method technically uses the schedule developer to create and link the IMS tasks. However, engineers and other stakeholders dictate the applicable information from the service guidance to the scheduler. Meetings with multiple engineers present can also be conducted for determining hand-offs between specialties and scheduling of joint tasks. There can be a lot of engineering time lost using this method but sometimes it is necessary to have the discussion in order for the scheduler to accurately capture the planned work.

Figure 6. “Engineering Interview” Method



All team members are inside the boundary of IMS development. IMS task creation and IMS task linking are performed by the schedule developer but the method requires the presence of the engineers.

There are pros and cons to having all this engineering and schedule development expertise in the same room at the same time. Pros include task naming conventions written by a schedule developer following explicit rules,

proper logic ties, and very little follow-up on the tasks created and linked. However, much time is consumed in dialog, typing, searching for predecessors and successors, and structuring. This method is often helpful and desirable in the final stages of IMS construction, but can be a very slow, expensive, and wasteful if used to develop the entire IMS.

2. NAVAIR Standard Work Packages

NAVAIR Standard Work Packages (SWP) contain work steps for processes that are repeated. For NAVAIR 4.0, SWPs are available to delineate the steps to produce or update artifacts for configuration items. (NAVAIR [4.0] 2015)

SWPs contain process steps that can be used to derive IMS tasking. The direct engineering effort used to create a SWP can be reused anytime IMS tasking is needed to create or update a particular artifact that is governed by the SWP. Since the focus of this thesis is concerned with engineering tasks, NAVAIR 4.0 SWPs are most significant. Additionally, SWPs (or their equivalents) are available for many processes (both inside and outside of NAVAIR) and can be used as sources for task creation.

THIS PAGE INTENTIONALLY LEFT BLANK

III. METHODOLOGY

A. CREATING AN ARCHITECTURE-BASED UTILITY FOR REPEATING TASK PLANNING

IMS development is an iterative process and a necessary activity. Overall, the degree to which engineers must engage in the development of the IMS depends on how knowledgeable the scheduler is concerning the technical work and dependencies. Ideally, engineers should be allowed, enabled, and encouraged to spend most of their time performing value-added activities that change the system while the administrative work of cost and schedule estimating is performed mostly independently by cost and schedule estimators. The scheduler can work fairly independently from TPWs if all information is present, but present methods use engineers to populate TPWs.

The use of TPWs on NAVAIR programs comes recommended and approved by NAVAIR 4.2.3 but requires a lot of direct engineering effort. A tool that automates creation of TPWs based on changes to systems is the primary objective of this thesis. A-BURTP was created to enable a scheduler to independently create and sequence the required tasks for an ECP with minimal disruption to program engineers. This allows engineers to spend their time on higher-value activities.

Figure 7 illustrates the methodology for A-BURTP. The methodology allocates TPW creation to the A-BURTP tool rather than to engineers or schedulers. As with the NAVAIR TPW process, linking of the IMS tasks within the scheduling tool is performed manually by the scheduler using information included in the TPW. A-BURTP was built in MS Access, creates TPWs in Excel format which are imported into MS Project.

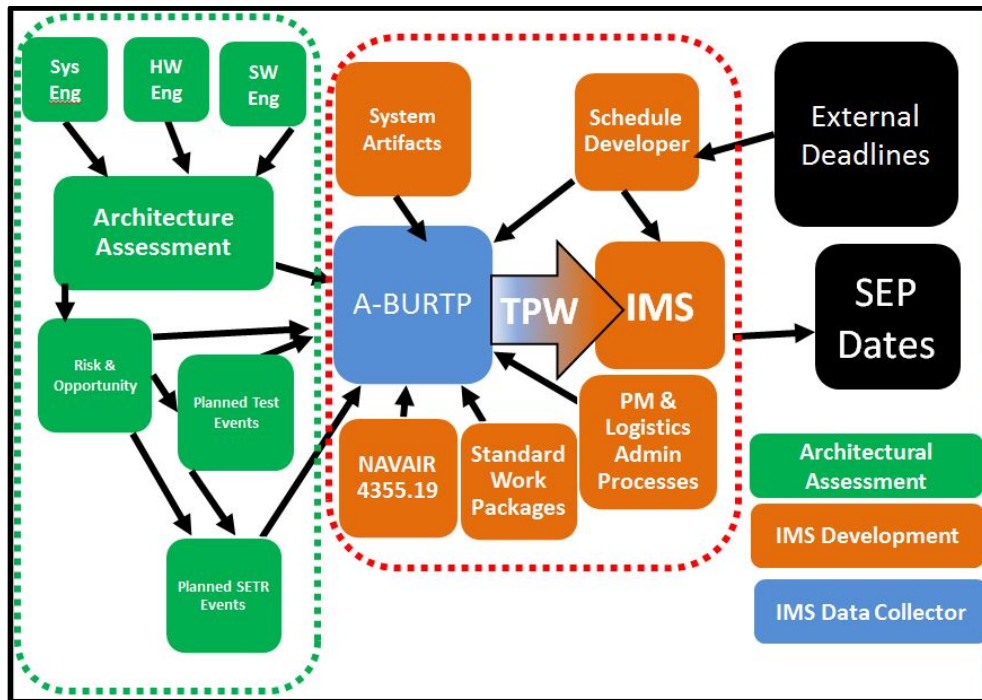
For this thesis, a simple Vitech CORE9 file was built (see Appendix A) to export a text file containing only system components, interfaces, and associated artifacts. These were used to populate data tables of A-BURTP. The architectural

assessment can take place within an MBSE tool such as Vitech CORE, but MBSE is not necessary. Any method engineers use to determine architecture and artifacts affected by an ECP, along with schedule risks and opportunities, can be used to populate the tables. Other A-BURTP tables contain standard work package work steps for maturing the artifacts and the NAVAIRINST 4355.19 SETR entry criteria provides collector milestones for the artifacts at required maturity levels.

A-BURTP requires engineering input from the system architectural assessment for an ECP. This means the engineers need to evaluate the changes needed for the ECP and determine the affected architecture items such as assemblies, components, interfaces, and software configuration items. This list of architecture items affected by the ECP is compared within A-BURTP and processed in queries that determine the associated tasks to be added to the TPW.

A master list of system architecture and artifacts, as well as the master list of NAVAIR 4355.19 SETR events and entry criteria, was loaded into A-BURTP. Then systems engineers just indicate the subset of SETR events planned for the ECP. With these few engineering inputs, A-BURTP can create the correct task list. Admin processes for program management and logistics technicians are held in other tables within A-BURTP and allow for the creation of those tasks.

Figure 7. Methodology for A-BURTP



Methodology for A-BURTP places engineers outside of the initial IMS development and focused on architectural assessments, determining testing requirements, and tailoring of reviews. A-BURTP automatically populates Task Planning Worksheets which are imported into the IMS and allow calculation of dates for the systems engineering plan.

B. OBJECTIVES

1. Minimize Direct Engineering Effort to Develop IMS Tasking

- A-BURTP shall not require engineers to describe actions contained in standard processes and instructions.
- A-BURTP shall not require engineers to describe task dependencies contained in standard processes and instructions.

2. Use Standard Output Format

- A-BURTP shall output task planning worksheets in MS Excel.
- A-BURTP shall output task planning worksheets that map directly into MS Project.

- c. A-BURTP shall output task planning worksheets that map to field definitions contained in the Appendix C data dictionary of the NAVAIR 4.2.3 Integrated Government Schedule Development and Maintenance Toolkit (NAVAIR [4.2.3] 2010, 32).

3. Create Task Names

A-BURTP shall construct unambiguous task names using an action/object naming convention (NDIA 2012).

4. Estimate Task Durations

- a. A-BURTP shall populate all task durations in whole-day integers.
- b. A-BURTP shall calculate initial three-point durations for SRA
 - Most-likely durations shall be based on complexity of architecture provided by engineers
 - Optimistic durations shall be based on most-likely durations and opportunity factors provided by engineers.
 - Pessimistic durations shall be based on most-likely durations and risk factors provided by engineers.

5. Assign Task Dependency

A-BURTP shall populate task dependency fields adequately to allow a schedule developer to independently link the tasks and produce a critical path with key milestone dates.

6. Populate WBS Field with System Architecture Data

A-BURTP shall apply WBS codes where defined or system architecture information if WBS is not defined.

7. Assign Engineering Review Type

A-BURTP shall assign review type to each task based on the review for which the task object is entry criteria.

8. Populate CDRL Field with Artifact Data

A-BURTP shall populate the “CDRL” field for each artifact task to describe what object is being produced or modified by the task action.

9. Populate Resource Names

A-BURTP shall populate the “Resource Names” text field (not the “Resource Name” MS Project resource field) based on task ownership and work step information.

10. Populate Basis of Estimate for Original Duration

A-BURTP shall populate the “Basis of Estimate for Original Duration” text field based on architecture complexity provided by engineers.

11. Populate Team or IPT Field

A-BURTP shall populate the “Team or IPT” field based on architecture assignment to team or IPT.

12. Populate Schedule Risk and Opportunity Field

A-BURTP shall populate the “Risk” text field based on risk and opportunity input from engineers.

C. FUNCTIONAL DECOMPOSITION OF TASK CREATION

A Vitech CORE9 file was constructed to decompose the functions of determining dates for the technical approach (SEP dates). The “Create Tasks” function was thoroughly decomposed and IDEF0 drawings were created to demonstrate the allocation of functions to components with the A-BURTP tool in the loop. Lower level functions within A-BURTP were also modeled in IDEF0 format. These drawings are discussed in this section.

1. “Calculate SEP Dates” Function

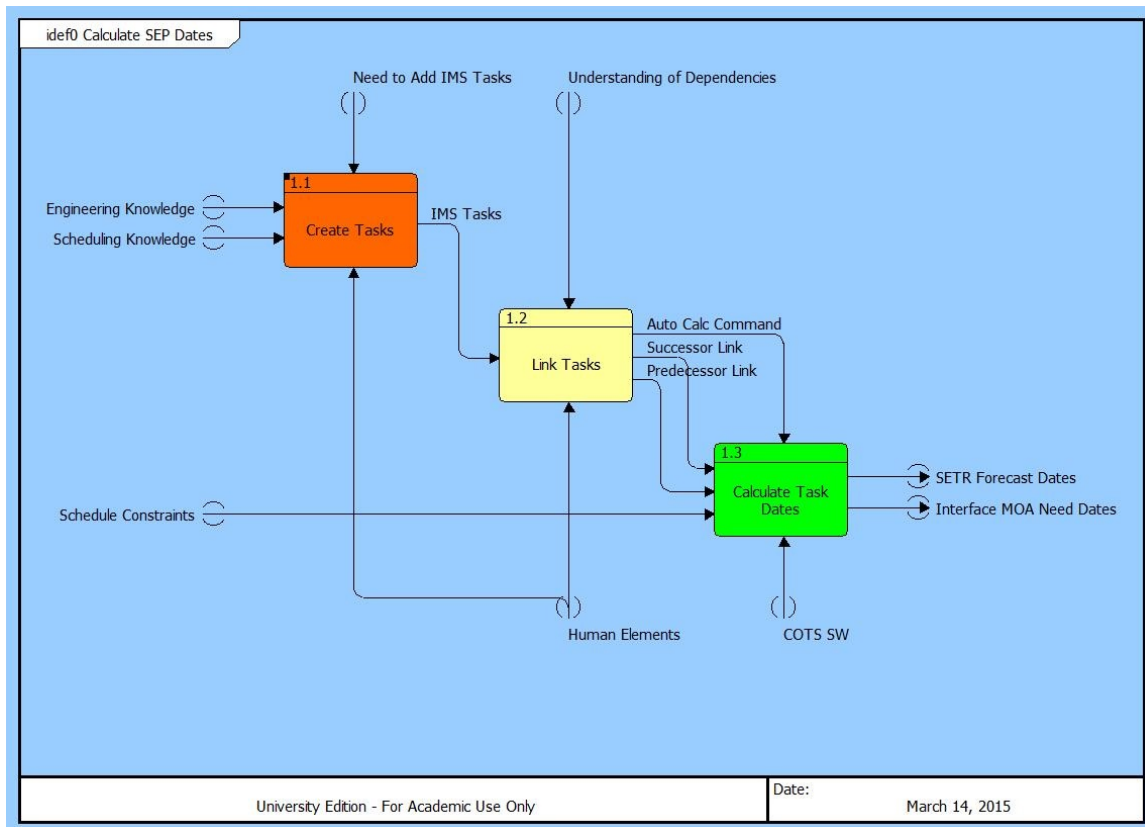
The SEP contains the technical approach and systems engineering activities while the IMS models a timeline for that work. For example, some of the IMS dates the SEP requires are forecast dates for SETR events and “Required by” dates for interface MOAs (OSD 2011a). In order for the IMS to calculate these dates, the task durations and dependencies must be known and input correctly into the scheduling software.

In Figure 8, the function of “Calculate SEP Dates” is decomposed into sub-functions which create tasks, place them in correct sequence, and then calculate projected dates for key milestones required in the SEP. Figure 8 names these three sub-functions “Create Tasks” (node 1.1), “Link Tasks” (node 1.2), and “Calculate Task Dates” (node 1.3).

a. *Create Tasks Function (Node 1.1)*

Node 1.1 (Create Tasks) is the foundational and most difficult function and is also the one with the most potential to compete for engineering resources. Node 1.1 is represented red in Figure 8 to indicate a high level of engineering involvement in the current methods. Node 1.1 is decomposed later in this section and lower level functions are allocated to A-BURTP. The inputs of “Engineering Knowledge” and “Scheduling Knowledge” are combined in node 1.1 by humans in order to create the tasks correctly.

Figure 8. Non-Automated IDEF0 “Calculate SEP Dates”



Non-automated IDEF0 “Calculate SEP Dates” highlights “1.1 Create Tasks” and “1.2 Link Tasks” functions presently performed by humans while “1.3 Calculate Task Dates” function is performed by commercial software.

b. Link Tasks Function (Node 1.2)

In Figure 8, the Link Tasks function (node 1.2) normally involves setting the predecessor and successor logic that describes the relationship between two tasks. Linking of tasks is sometimes performed by engineers and sometimes by a scheduler. This function remains a manual effort by humans and therefore no further decomposition of this function was performed in this thesis. Node 1.2 is yellow because linking tasks is allocated to humans even when the A-BURTP tool is used for the Create Tasks function (node 1.1).

c. Calculate Task Dates Function (Node 1.3)

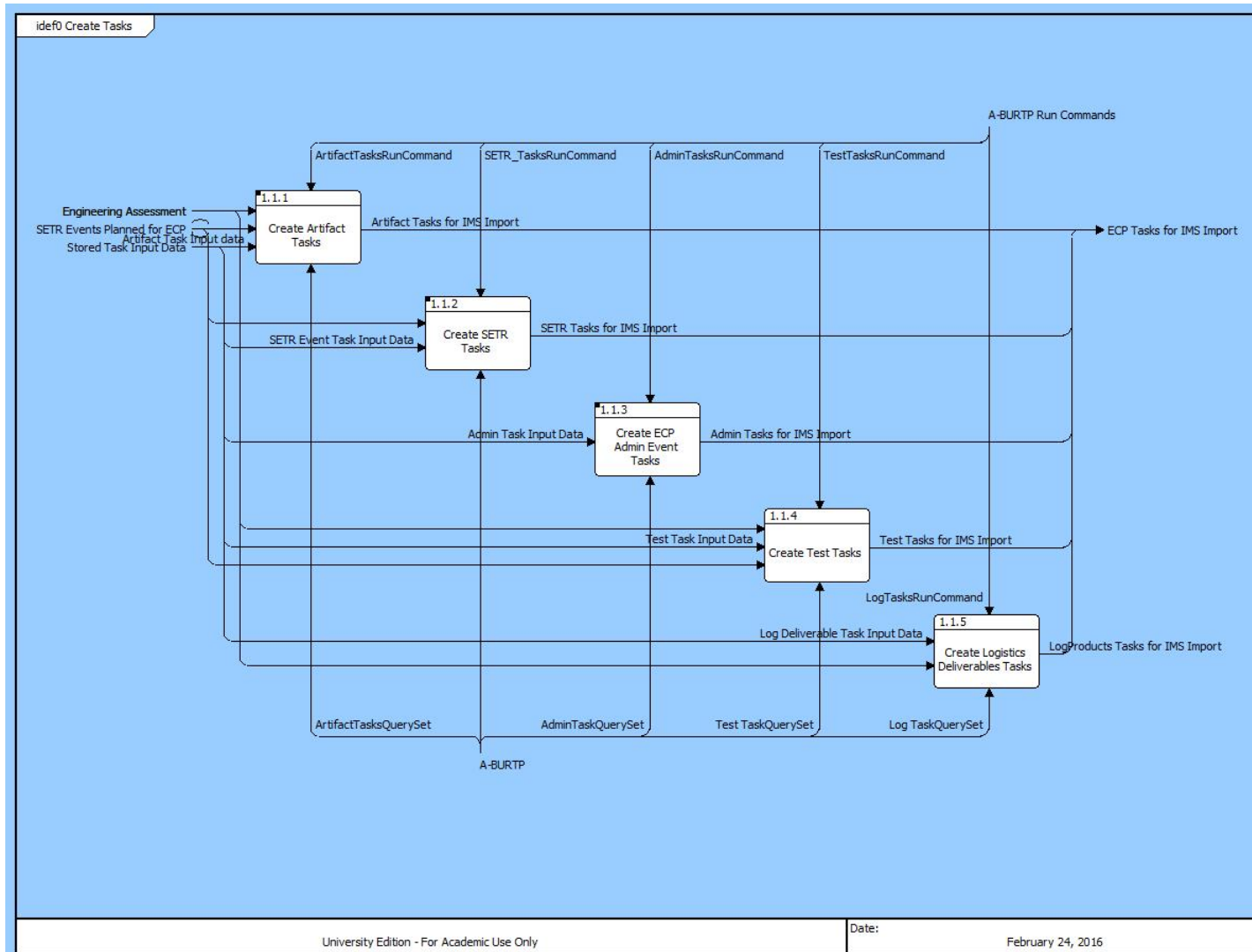
The function of Calculate Task Dates (node 1.3) is allocated to COTS scheduling software. MS Project was used in the Results section demonstration because it is available on the Navy Marine Corps Intranet machines. All off-the-shelf scheduling tools follow scheduling standard practices and the results of calculation reflect the aggregate schedule based on task durations, task dependencies, and task constraints. If the prior two functions are performed correctly, the IMS can be used to provide forecast dates for technical objectives in the SEP.

Node 1.3 is represented green in the IDEF0 in Figure 8 because no engineering effort is required. No further decomposition of node 1.3 was performed in this thesis.

2. Decomposition of Create Tasks Function, Automated

The Create Tasks function from Figure 8 is further decomposed to an intermediate level in Figure 9 and performed by A-BURTP. While there is still need for input from outside of A-BURTP, much of the information required to create tasks comes from data stored inside. Only the engineering assessment and the planned SETR events are needed from the engineers. The engineering assessment must include the affected system components, interfaces and other architecture items along with their complexity and risk bands. A-BURTP derives the remaining information from stored data.

Figure 9. Create Tasks Function Using A-BURTP



The Create Tasks function is decomposed into various task types needed for the ECP.

Creating tasks is a very involved function as IMS tasks contain many attributes. Different types of tasks have different attributes and the sub-functions for creating different task types are also different.

The “Engineering Knowledge” that was an input in Figure 8 is now split between information stored in A-BURTP and outputs from an engineering assessment of the system changes. The stored information came from a mixture of sources. Information from engineering directives, instructions, and local processes were stored in A-BURTP tables. Likewise, system artifact document names and the associated system components, assemblies, and interfaces they describe were stored in A-BURTP.

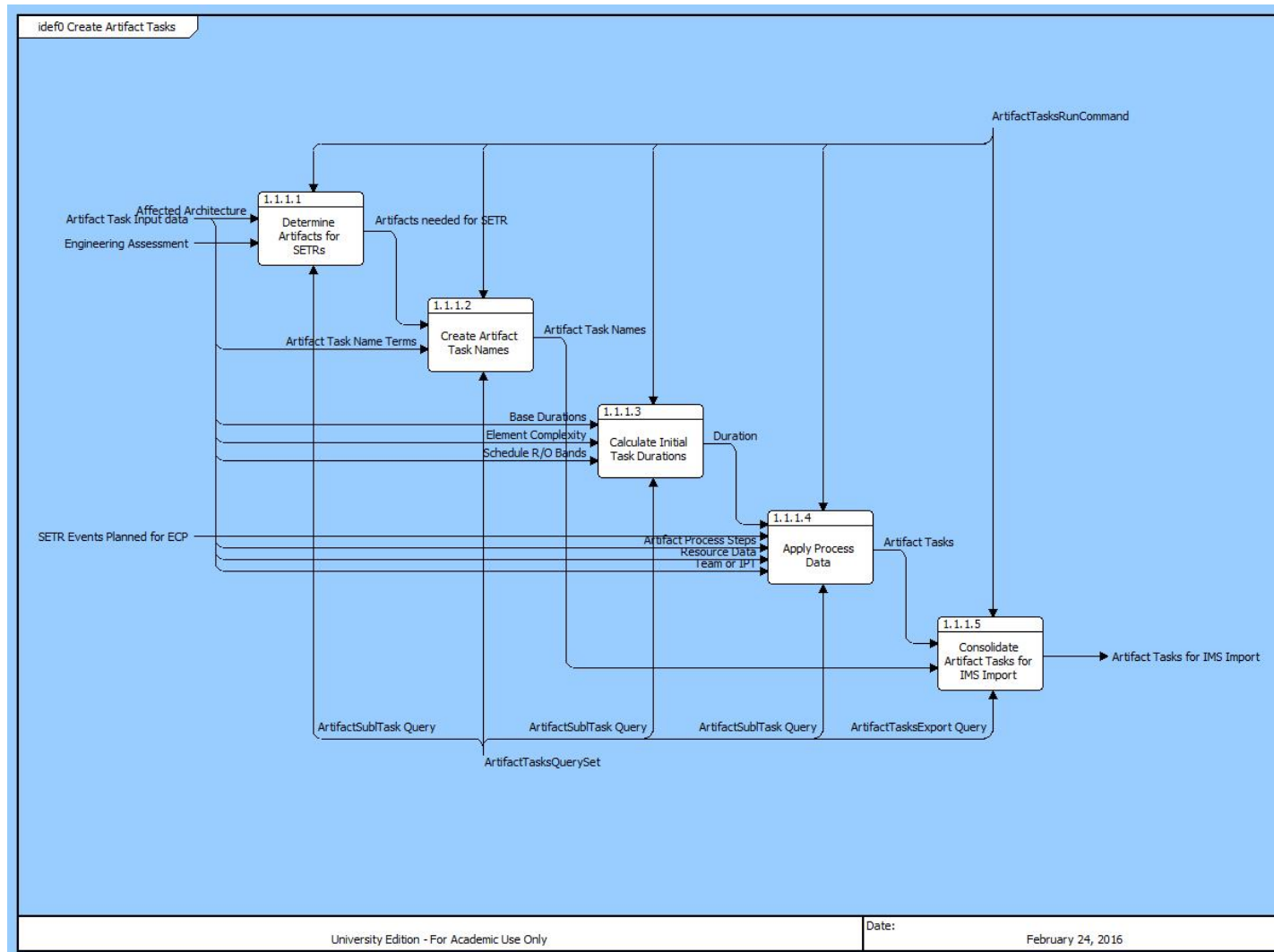
The information from engineering rigor such as ECP architectural assessments and decisions on which engineering reviews will be conducted for each ECP comes from the engineers and is shown as external inputs. The other inputs needed from engineers are the complexity of the architecture and the risk/opportunity factors. These are explained in the Create Artifact Task Function section. The “Scheduling Knowledge” that was an input in Figure 8 is now programmed into the A-BURTP query sets. As the queries are applied correctly to the engineering knowledge, many attributes of each task type were determined.

The “create tasks” function is allocated to humans in the NAVAIR TPW process and in Figure 8. When the A-BURTP tool is used, some functionality is allocated to the tool and some functionality remains allocated to humans. Table 10 in the Analysis of Success in Meeting Objectives section contains a comparison of input sources for the two methods. Figures 10 through 12 illustrate the functions performed by queries within A-BURT. These include Create Artifact Tasks (Figure 10), Create SETR Tasks (Figure 11), and Create ECP Admin Event Tasks (Figure 12).

a. *Create Artifact Tasks Function*

Figure 10 decomposes the function of Create Artifact Tasks (node 1.1.1) as performed by A-BURTP. The sub-functions for Figure 10 are performed by a query set pulling data stored within A-BURTP.

Figure 10. Create Artifact Tasks Function



Decomposed to show how many task attributes are collected and applied to create the automated TPW.

(1) Determine Artifacts for SETRs (Node 1.1.1.1)

This function uses the input from the engineers concerning the SETR events and affected architecture items applicable to the ECP. Additionally, the SETR entry criteria are pulled from the NAVAIR 4355.19 data stored within A-BURTP. The queries then use the affected architecture items to determine the affected artifacts needed for the required SETR events. These artifacts become the objects in the action/object task name. Table 3, Table 7, and Table 8 in the Data Sources and Storage section of this thesis provide more detailed information concerning 4355.19 data.

(2) Create Artifact Task Names (Node 1.1.1.2)

As the PASEG calls for action/object task naming conventions, it is an objective that the task name be unambiguous (NDIA 2012, 40). The Artifact Task Query Set creates properly named summary and sub tasks. The task names are concatenated to include the architecture item, the artifact, and the acronym on summary tasks, and the action, architecture item, acronym, and maturity level on subtasks. The maturity level comes from the process and process step information from the SWP work steps. Table 5 and Table 6 in the Data Sources and Storage section of this thesis provide more detailed information concerning process steps.

(3) Calculate Initial Task Duration (Node 1.1.1.3)

Three-point task durations are required by the PASEG (NDIA 2012, 148) in order to run Monte Carlo simulations which ultimately feed into the SRA required by the SEP (OSD 2011a). Nominal base durations are included in the process steps and these are multiplied by an architecture complexity factor to produce the most likely duration. The most likely duration is multiplied by the risk and opportunity factors to calculate the pessimistic and optimistic durations. All of these calculations are rounded up to the nearest integer to eliminate fractional

day results. Table 3 and Table 6 in the Data Sources and Storage section of this thesis provide more detailed information concerning duration data.

In this thesis, complexity and risk/opportunity inputs come from engineers. The ability to derive these factors from architecture is discussed in Topics for Future Research and credited to Tan's thesis work (Tan 2012).

(4) Apply Process Data (Node 1.1.1.4)

Process data helps fulfill the Task Dependency objective. This data includes the numerical process steps from the SWP work steps that allow the schedule developer to link the tasks in sequence without assistance from the engineers. Also, the attribute Team or IPT allows grouping and filtering by task owner. Nominal resource figures are applied that the team leaders can adjust later in the IMS development process if necessary.

(5) Consolidate Artifact Tasks for IMS Import (Node 1.1.1.5)

The final function gathers all the data, consolidates the tasks, and appends them into an automated TPW. The outputs from all other functions in Figure 10 are inputs to this function. Of note is the SETR predecessor data coming from the Determine Artifacts for SETRs. This information comes from tables within the tool that contain NAVAIRINST 4355.19 SETR entry criteria and also helps fulfill the Task Dependency objective. A-BURTP assigns a review type to each artifact task to enable the schedule developer to link them once inside the IMS.

b. Create SETR Tasks Function

The sub functions for creating SETR tasks (node 1.1.2 in Figure 9) are different from those for the artifact tasks in which specific work products are created, modified, reviewed, and approved. Figure 11 illustrates the sub-functions of creating SETR tasks for import and appending them to the TPW. SETR event tasks start with the systems engineer, hand off to the team, collect back to the systems engineer, and so forth and are done specifically to support

SETR review events. The main difference in this function from the Create Artifacts Tasks function is that it includes union query to enable situations with one predecessor task and many successors, and also situations with one successor and many predecessors. The systems engineer must provide the planned SETR strategy for the ECP to A-BURTP to create the correct tasks and dependency data.

[illegible]

48

(1) Collect SETR Key Tasks (node 1.1.2.1)

Key tasks include tasks that are not owned by any specific area of the team. These are the team reviews, the dry-run, the “ready for” milestone, the margin task, and the conduct event task. These are gathered from a separate query because they are not owned by any particular team.

(2) Create SETR SE Tasks (node 1.1.2.2)

SETR SE Tasks includes those tasks that are only systems engineering responsibilities such as tailoring the SETR checklist and consolidating team input for SETRs. Only one each of these tasks is created per SETR event.

(3) Create SETR Team Tasks (node 1.1.2.3)

SETR Team Tasks includes those tasks that each area of the team performs individually. These include slides by each area of the team, requests for action by each area of the team, and review of SETR minutes by each area of the team.

(4) Consolidate SETR Tasks for IMS Import (node 1.1.2.4)

No three-point estimates are applied to the SETR tasks because the risk and opportunity are captured in the artifact tasks. The nominal durations come from the process steps but no complexity or risk/opportunity factors are applied. This query appends the SETR Key tasks, SETR SE tasks, and the SETR Team tasks to the TPW. Table 9 in the Data Sources and Storage section of this thesis contains the SETR task specifics.

c. Create ECP Admin Event Tasks Function

The sub functions for creating ECP admin events (node 1.1.3 in Figure 9) depends on whether the ECP is a Class 1 or Class 2. Both require a final configuration control board (CCB) approval, but Class 2 ECPs do not require a decision memorandum step or the predecessor tasking (NAVAIR [PMA-251]

2013). Figure 12 illustrates the sub-functions of creating ECP admin event tasks for import.

```

    usecaseDiagram
        participant Input as Admin Task Input Data
        participant U1 as 1.1.3.1 Collect ECP Admin Key Event Tasks
        participant U2 as 1.1.3.2 Collect ECP Admin Logistics Tasks
        participant U3 as 1.1.3.3 Collect ECP Admin PM Tasks
        participant U4 as 1.1.3.4 Collect ECP Admin Team Tasks
        participant U5 as 1.1.3.5 Consolidate ECP Admin Tasks for IMS Import
        participant Output as Admin Tasks for IMS Import

        Input --> U1 : Key Admin Steps
        Input --> U1 : ECP Class
        Input --> U2 : LOG Admin Steps
        Input --> U2 : ECP Class
        Input --> U3 : PM Admin Steps
        Input --> U3 : ECP Class
        Input --> U4 : Team Admin Steps
        Input --> U4 : ECP Class
        Input --> U5 : CollectAdminKeyTasks Query
        Input --> U5 : CollectAdminLogisticsTasks Query
        Input --> U5 : CollectAdminPMTasks Query
        Input --> U5 : CollectAdminTeamTasks Query
        Input --> U5 : AdminTaskQuerySet

        U1 --> U2 : Key Admin Tasks
        U2 --> U3 : LOG Admin Tasks
        U3 --> U4 : PM Admin Tasks
        U4 --> U5 : Team Admin Tasks

        U1 --> U5 : AdminTasksRunCommand
        U2 --> U5 : AdminTasksRunCommand
        U3 --> U5 : AdminTasksRunCommand
        U4 --> U5 : AdminTasksRunCommand
        U5 --> Output
    
```

The diagram illustrates the process of creating ECP Admin Event Tasks. It starts with 'Admin Task Input Data' which feeds into four parallel collection steps (1.1.3.1 to 1.1.3.4). Each step receives 'Key Admin Steps' and 'ECP Class' as inputs. The outputs of these steps are 'Key Admin Tasks', 'LOG Admin Tasks', 'PM Admin Tasks', and 'Team Admin Tasks' respectively. These tasks are then consolidated in step 1.1.3.5, which also receives 'AdminTaskQuerySet' and 'AdminTasksRunCommand' as inputs. The final output is 'Admin Tasks for IMS Import'.

51

(1) Collect ECP Admin Key Event Tasks (node 1.1.3.1)

Key tasks include a pre-CCB, the “ready for CCB” milestone, the margin task, and the conduct CCB task.

(2) Collect ECP Admin Logistics Tasks (node 1.1.3.2)

These tasks are logistics only.

(3) Collect ECP Admin PM Tasks (node 1.1.3.3)

These tasks steps are for PM only

(4) Collect ECP Admin Team Tasks (node 1.1.3.4)

These tasks are common to all team members

(5) Consolidate ECP Admin Tasks for IMS Import (node 1.1.3.5)

Admin tasks also include external tasks for waiting periods that must be monitored while the ECP package is out for review. No three-point estimates are applied to the ECP admin tasks because the risk and opportunity are captured in the artifact tasks.

3. Summary of Functional Decomposition

The Create Tasks function (node 1.1 in Figure 8) was decomposed to understand the creation of three types of tasks required for IMS development. Artifact tasks (Figure 10), SETR tasks (Figure 11), and Admin tasks (Figure 12) require different sub-functions to create. Test event tasks, and logistics deliverable tasks are some additional types of tasks shown in Figure 9 that were not within scope of this thesis but could be decomposed and included in future work.

D. DATA SOURCES AND STORAGE

1. Introduction to Data Sources and Storage

This section explains how various data sources are gathered and stored in A-BURTP to enable queries that produce fully populated TPWs.

2. Background

A very simplistic Vitech CORE9 MBSE file was created to model architecture of a fictional system in order to export a text table for the MBSE “documented by” relationship. This provided a few components and interfaces and associated artifacts to be used in A-BURTP to demonstrate the ability to create TPWs for import into the IMS file. The MBSE file and associated drawings are discussed in Appendix A and details are not discussed here. A-BURTP can also be populated by the scheduler with information from the SEP and its reference documents if a SEP is available.

3. Data Storage in A-BURTP Tables

A-BURTP requires input from several sources to create the TPWs. Much of this data collected serves double duty to satisfy the requirements of section 4 of the SEP as discussed in the literature review of the OSD Systems Engineering Plan Outline (OSD 2011a). Information required for A-BURTP are engineering review entry criteria, configuration item documentation, change processes, and configuration management processes. These information items are gathered and the data is distributed throughout tables within A-BURTP. If the SEP section 4 is in correct form, it can be used for this purpose.

a. Input Elements Table

Table 3 is the Input Elements table within A-BURTP. The “element” field is populated with system architecture data concerning configuration items and interfaces. Table 3 also contains the Element type, the complexity level, and the Risk Opportunity Band. In this table, engineering input is also needed to classify the schedule risk/opportunity band and the complexity of each configuration item.

In this sample data, all elements are checked “affected” indicating that tasks concerning that element need to be created for this example ECP.

The data in Table 3 can be populated once and held. Only the “affected” field needs to be adjusted for a new ECP. The other fields can remain the same if nothing changes.

Table 3. “Input Elements” Table from A-BURTP

Input Elements				
Element Type	Element	Affected	Risk Opportunity Band	Complexity
HWCI	UPS	True	Low/High	Low
HWCI	Network Switch	True	Medium/Medium	Medium
CSCI	App XYZ	True	Medium/Low	Medium
CSCI	App ABC	True	High/Low	High
CSCI	OS	True	Medium/Medium	Medium
HWCI	Cooling Fan	True	Low/Medium	Low
HWCI	Classified Server	True	Medium/Medium	Low
Assembly	Rack Assembly	True	Low/High	Medium
SystemLevel	System A	True	High/Low	High
Interface	System A/System B	True	Medium/Low	High
Interface	System A/System C	True	High/Low	High
Interface	System A/CVN	True	Low/Medium	High

Sample data with complexity and risk/opportunity factors assigned.

b. Artifacts Table

Table 4 is the A-BURTP “Artifacts” table. The “Process Number” field is used to determine the actions necessary for each artifact object. A-BURTP uses the processes in Table 5 as a drop-down list for the Process Number field in Table 4. This prevents errors when populating Table 4. Acronyms are standardized to those used as SETR entry criteria.

Table 4. “Artifacts” Table from A-BURTP

Artifacts			
Acronym	Artifact	Process Number	Team or IPT
IRS	Interface Requirement Specification	10	SE
SRS	System Requirement Specification	9	SE
SwRS	Software Requirements Specification	9	SW
IDD	Interface Design Document	9	HW
HCS	Hardware Component Specification	9	HW
STP	Software Test Plan	9	SW
DRG	Drawing	9	HW
BWD	Block Wiring Diagram	9	HW
ICD	Interface Control Drawing	9	HW

Classes of artifacts mapped to acronyms, team ownership, and local processes.

c. Processes Table

The A-BURTP “Processes” Table shown in Table 5 contains the processes available for assignment to each artifact in Table 5 as well as processes for SETR objects in Table 9. A-BURTP used simulated data but, ideally, these processes would be actual NAVAIR SWPs or other documented local processes.

Table 5. “Processes” Table from A-BURTP

Processes	
ProcessNumber	ProcessName
1	Submit for an ECP Decision Memorandum
2	Create Planning Tasks
3	SETR Checklist
4	SETR Slides
5	SETR Minutes
6	SETR Key Tasks
7	ECP Closeout DCCB
8	Create a New Design Document
9	Change an Existing Internal Design Document
10	Create a New External Interface Requirements Specification
11	Change an Existing External Interface Requirements Specification
12	Down-Select a COTS item
13	Code and Unit Test Software
14	ECP Integration Test

Sample local processes available to map to system artifacts.

d. Process Steps Table

Table 6 displays the “Process Steps” table from A-BURTP. The table contains the steps for each process in Table 5 and delineates the engineering process steps through which system artifacts are matured. This information can be obtained by breaking out the work steps found in NAVAIR SWPs. This table contains low level details and requires some additional work to research and input into A-BURTP.

Step “0” designates summary tasks which receive a different naming convention than subtasks. The “Action” column contains the verbs to be concatenated into the subtask names where the process step is greater than 0. Table 6 is filtered for clarity to show only the steps for processes 9 and 10 (from Table 5). Base duration is used in four separate task attribute calculations.

Table 6. “Process Steps” Table from A-BURTP

Process Steps					
Process	Process Step	Action	Maturity	Base Duration	Resource Units
9	0	Summarytask	Final	0	0.00%
9	1	Write	Preliminary	10	50.00%
9	2	Review	Preliminary	5	50.00%
9	3	Revise	Preliminary	2	50.00%
9	4	Finalize	Preliminary	5	50.00%
9	5	Sign	Final	5	50.00%
10	0	Summarytask	Final	0	0.00%
10	1	Write	Preliminary	10	50.00%
10	2	Review	Preliminary	5	50.00%
10	3	(SVT) External Review of	Preliminary	15	0.00%
10	4	Receive	Preliminary	0	0.00%
10	5	Revise	Preliminary	5	50.00%
10	6	Finalize	Preliminary	5	50.00%
10	7	Sign	Final	5	50.00%

Data filtered to show steps from engineering processes 9 and 10 includes base durations and resources to mature artifacts.

e. Input Select Reviews Table

The A-BURTP “Input Select Reviews” data shown in Table 7 requires input from the user to select the reviews planned for the ECP. This information comes from the systems engineer and is input by the scheduler. Table 7 has SRR, PDR, CDR, and TRR selected as “True” so A-BURTP will generate tasking for those reviews.

Table 7. “Input Select Reviews” Table from A-BURTP

Input Select Reviews		
Sort	ECP Reviews	Planned
0	DM	False
1	SRR	True
2	SRR-PDR	False
3	SRR-PDR-CDR	False
4	PDR	True
5	PDR-CDR	False
6	PDR-CDR-TRR	False
7	CDR	True
8	CDR-TRR	False
9	TRR	True

ECP technical reviews set to “True” by the schedule developer using system engineering input.

f. Artifact Maturity Review Table

Table 8 contains sample data but is not populated with all artifacts and all reviews. Complete population of this table would be a direct engineering effort, but once populated, would not need to be repeated. If it was available from the program SEP, the schedule developer could enter the data and further reduce the burden on engineers.

Combined reviews in Table 7 are accommodated by linking artifacts to the latest SETR in the combination. In Table 8, artifacts due at SRR-I and SRR-II are associated to SRR in Table 8.

Table 8. “Artifact Maturity Review” from A-BURTP

Artifact Maturity Review			
Artifact	Maturity	EntryToReview	Resource
Interface Design Document	Preliminary	SRR-II	HWME
Interface Requirement Specification	Final	SRR-II	Systems
Software Requirements Specification	Final	PDR	SWDEV
Software Requirements Specification	Preliminary	SRR-II	SWLead
Interface Design Document	Final	PDR	HWLead
System Requirement Specification	Final	SRR-I	Systems
Interface Requirement Specification	Preliminary	SRR-I	Systems
Drawing	Preliminary	PDR	HWME
Drawing	Final	CDR	HWLead
Block Wiring Diagram	Preliminary	PDR	HWEE
Block Wiring Diagram	Final	CDR	HWLead
Interface Control Drawing	Preliminary	PDR	HWNE
Interface Control Drawing	Final	CDR	HWLead

Artifact maturity levels are mapped to review entry criteria and assigned to resources.

SETR events require team tasks and SE-specific tasks. These actions and objects are handled in a separate table from the system artifacts. Table 9 shows some steps exclude SE, some are SE only, and some are team tasks. Fields containing duration, process, and process step data also exist in A-BURTP “SETR Object Steps” table but are not shown in Table 9 due to size constraints.

Table 9. “SETR Object Steps” Table from A-BURTP

SETR Object Steps				
Action	Maturity	SETR Object	SE Only	Exclude SE
		Common Team Tasks	False	False
Tailor		Checklist	True	False
Work	Tailored	Checklist	False	False
Populate database with		Checklist data	True	False
Create and Send to Team		Slide Template	True	False
Populate	Draft	Slides for Team Review	False	False
Consolidate and Send to Team	Draft	Slide Deck for Team Review	True	False
Update	Draft	Slides from Team Review	False	False
Consolidate and Send to TRB	Semi-Final	Slide Deck for Dry Run	True	False
Incorporate Comments on	Semi-Final	Slides from Dry Run	False	False
Finalize		Slide Deck	True	False
Create	Pre-Written	RFAs	True	False
Work		RFAs	False	False
Verify and Request Closure of		RFAs	True	False
(SVT) Chair Approval to Close		RFAs	True	False
Write and send to team	Draft	Minutes Report	True	False
Provide Input from	Draft	Minutes Report	False	True
Consolidate team comments and submit to Chair	Draft	Minutes Report	True	False
(SVT) Chair Review of	Draft	Minutes Report	True	False
Incorporate Comments and Submit	Final	Minutes Report	True	False
		SE-Specific Tasks	True	False

Steps for SETR events are designated “SE Only” and “Exclude SE” where applicable.

E. ANALYSIS OF SUCCESS IN MEETING OBJECTIVES

This section analyzes the success of A-BURTP in performing the stated objectives. The ease of use, quality of TPW data, and engineering input efforts are discussed.

1. Minimize Direct Engineering Effort to Develop IMS Tasking

While actual time studies were not performed on manual TPW creation, Table 10 illustrates the reduction in required engineering input when A-BURTP was used. Further, A-BURTP created 320 tasks using only a few minutes of engineering input time. While it took engineering time to populate some of the A-BURTP tables, this was a one-time effort that can be reused for future ECPs

- a. A-BURTP shall not require engineers to describe actions contained in standard processes and instructions.

Table 10. Task Planning Worksheet Data Sources

TPW Data	TPW Field Population Source	
	Current Processes	A-BURTP
Planned Reviews	Engineers	Engineers
Architecture affected	Engineers	Engineers
Artifacts affected	Engineers	Derived
Task Duration	Engineers	Derived
Task Dependencies	Engineers	Derived
Task Name	Engineers	Derived
Task Resources	Engineers	Derived
Task Basis of Estimate	Engineers	Engineers
Team or IPT	Engineers	Derived
Risk Information	Engineers	Engineers
Optimistic Duration	Engineers	Derived
Pessimistic Duration	Engineers	Derived

Data input sources compared between A-BURTP and current processes demonstrates reduction in engineering effort required.

Simulated SWPs were used for artifact task process steps to test this objective. A-BURTP does not require engineers to describe actions contained in standard processes and instructions and met this objective using simulated data. Actual SWPs or other organizational process documentation would require audit and data normalization to initially load standard process steps, resources, and durations into A-BURTP.

- b. A-BURTP shall not require engineers to describe task dependencies contained in standard processes and instructions.

A-BURTP met this objective when tested with simulated SWP data and simulated artifacts as entry criteria to actual SETR events. Dependencies within SWPs were properly transferred to the TPW. Dependencies between artifacts and SETR entry criteria were also transferred properly to the TPW. Engineers were not required to describe dependencies contained in standard processes and instructions. As above, actual SWPs or other organizational process documentation would require audit and data normalization to initially load standard process steps, resources, and durations into A-BURTP. This is work that could be done once and not required again.

2. Use Standard Output Format

- a. A-BURTP shall output task planning worksheets in MS Excel.
- b. A-BURTP shall output task planning worksheets that map directly into MS Project.
- c. A-BURTP shall output task planning worksheets that map to field definitions contained in the Appendix C data dictionary of the NAVAIR 4.2.3 Integrated Government Schedule Development and Maintenance Toolkit (NAVAIR [4.2.3] 2010).

A-BURTP produced 320 tasks in Excel format for direct import to MS Project (MSP). Table 11 displays one task produced by A-BURTP along with the NAVAIR 4.2.3 Integrated Government Schedule Toolkit Appendix C Data Dictionary and associated MSP Field. All objectives for output format are met.

Table 11. Microsoft Project Standard Fields Assigned to NAVAIR 4.2.3 Data Dictionary Are Populated by A-BURTP

MSP Field	Appendix C Data	Automated TPW Result
Duration	Duration	20
Duration1	Optimistic Duration	18
Duration2	Pessimistic Duration	40
Duration3	Most-Likely Duration	20
Flag10	PMA Key Milestone	FALSE
Flag18	Milestone Status	FALSE
Name	Name	Write Preliminary System A/System C IRS
Number17	ImportSort	48
Number18	Test Asset	0
Number19	Process Number	10
Number20	Process Step	1
Text3	WBS_	System A/System C Interface
Text6	Review Type	SRR
Text7	CDRL Number	System A/System C Interface Requirement Specification
Text12	Resource Name	Systems[50%]
Text15	Dependencies	
Text22	Task BOE	System A/System C Interface complexity rated 'High' during initial interview; Complexity Factor = 2
Text24	Team or IPT Name	SE
Text25	Risk	System A/System C Interface Schedule Risk/Op Band rated High/Low; RiskFactor = 2; OppFactor = -0.1

Data for one task exported from A-BURTP detail of task attributes to be imported into schedule.

3. Create Task Names

A-BURTP shall construct unambiguous task names using an action/object naming convention (NDIA 2012).

A-BURTP concatenates artifact task names from a process step action and a direct object. For example, as shown in Table 11, the task name is “Write Preliminary System A/ System C IRS.” The action is “Write” and the direct object is constructed from the artifact maturity level (Preliminary), architecture name (System A/System C) and the artifact acronym (IRS).

Task names are constructed through concatenation of actions, objects, and architectural information where applicable. Concise task names constructed with the repeating phrase structure allow the team to quickly grasp the scope of the task. Table 12 is filtered on one SwRS to show the summary and subtasks in

that group. The column headings are default field names in MS Project and allow grouping, sorting, and filtering once imported into the IMS module.

Also shown are Text3, which contains the WBS/system architecture information, and Number19 and Number20 fields which are the process and process step fields that aid the scheduler in making logic links. These are also discussed in the Task Dependency section.

Table 12. Task Name, Task Dependency, and System Architecture Data from A-BURTP

xExportFormat			
Name	Number19	Number20	Text3
App ABC Software Requirements Specification (SwRS)	9	0	App ABC CSCI
Write Preliminary App ABC SwRS	9	1	App ABC CSCI
Review Preliminary App ABC SwRS	9	2	App ABC CSCI
Revise Preliminary App ABC SwRS	9	3	App ABC CSCI
Finalize Preliminary App ABC SwRS	9	4	App ABC CSCI
Sign Final App ABC SwRS	9	5	App ABC CSCI

Unambiguous action/object task naming conventions are demonstrated.

This requirement is met because the action/object naming convention is satisfied and A-BURTP uses unambiguous descriptive noun phrases for direct objects.

4. Estimate Task Durations

- a. A-BURTP shall populate all task durations in whole-day integers.
- b. A-BURTP shall calculate initial three-point durations for SRA.
 - Most-likely durations shall be based on complexity of architecture provided by engineers.
 - Optimistic durations shall be based on most-likely durations and opportunity factors provided by engineers.

- Pessimistic durations shall be based on most-likely durations and risk factors provided by engineers.

Table 13 shows the calculation formula for each duration field required on the TPW. The Value column data comes from the sample in Table 9. This shows how the four duration fields required by NAVAIR 4.2.3 are populated by A-BURTP. Base duration, architecture complexity, and risk/opportunity factors are included in the calculations and engineers must evaluate complexity, risk, and opportunity for each architecture item being modified as high, medium, or low. All combinations of risk, complexity, and opportunity and the high, medium, low factors are possible. Factors are not adjusted but the TPW or the IMS durations can be adjusted if the engineer wishes to fine tune the results.

Table 13. A-BURTP Duration Formulas

MSP Field	Appendix C Data	Formula	Value
Duration	Duration	Round([10Complexity]![Complexity_factor]*[Base_Duration],0)	20
Duration1	Optomistic Duration	IIf([Base_Duration]>0,Int([10Complexity]![Complexity_factor]*[Base_Duration]+([10Risk_Opportunity]![OpportunityFactor]*[Base_Duration]*[Complexity_factor])),0)	18
Duration2	Pessimistic Duration	IIf([Base_Duration]>0,Int([10Complexity]![Complexity_factor]*[Base_Duration]+([10Risk_Opportunity]![RiskFactor]*[Base_Duration])),0)	40
Duration3	Most-Likely Duration	Round([10Complexity]![Complexity_factor]*[Base_Duration],0)	20

A-BURTP duration formulas calculate initial three-point task durations in whole-day integers based on complexity and risk/opportunity inputs.

5. Assign Task Dependency

A-BURTP shall populate task dependency fields adequately to allow a schedule developer to independently link the tasks and produce a critical path with key milestone dates.

The Link Tasks function from Figure 8 (node 1.2) is still performed manually by the schedule developer but now without engineering assistance. Importing, filtering, grouping, sorting, and linking are operations familiar to users of MS Project. Finding and linking the predecessor and successor tasks is also

familiar and often takes time to hunt through the schedule to find the exact tasks to link.

The Dependency field in Table 9 is not populated by A-BURTP. The TPW can be sent to engineers to add dependency information if necessary. However, the dependency requirement is met through a combination of other fields. The Review Type, Process, and Process Step fields are used in combination to provide very concise dependency information. Figure 13 illustrates steps (a) through (e) to demonstrate the ease of linking tasks imported from A-BURTP.

Figure 13. IMS Tasks Imported from A-BURTP

Name	Review Type	ID	Process Number	Process Step	Predecessors	Successors	Team or IPT
Tailor SRR Checklist	SRR	140	1	1	Automatically Populated	34,93,154,245,274,303	SE
Work HW Tailored SRR Checklist	SRR	34	1	2	140	141	HW
Work SW Tailored SRR Checklist	SRR	93	1	2	140	141	SW
Work SE Tailored SRR Checklist	SRR	154	1	2	140	141	SE
Work IA Tailored SRR Checklist	SRR	245	1	2	140	141	IA
Work LOG Tailored SRR Checklist	SRR	274	1	2	140	141	LOG
Work PM Tailored SRR Checklist	SRR	303	1	2	140	141	PM
Populate database with SRR Checklist data	SRR	141	1	3	34,93,154,245,274,303		SE

IMS tasks imported from A-BURTP are manually linked using common schedule developer skills.

Step (a) is filtering on the Review Type and Process Number fields to cut the list down to the SE-specific tasks and the team tasks involved in Process 1. Step (b) is sorting by Process Step which puts the tasks in chronological order.

In this example, there is one predecessor that must be applied to many successors. Step (c) is copying the ID of the task with Process Step 1 (“Tailor SRR Checklist”) and pasting it in the predecessor field of the first task with Process Step 2 (“Work HW Tailored Checklist”). This creates the first link.

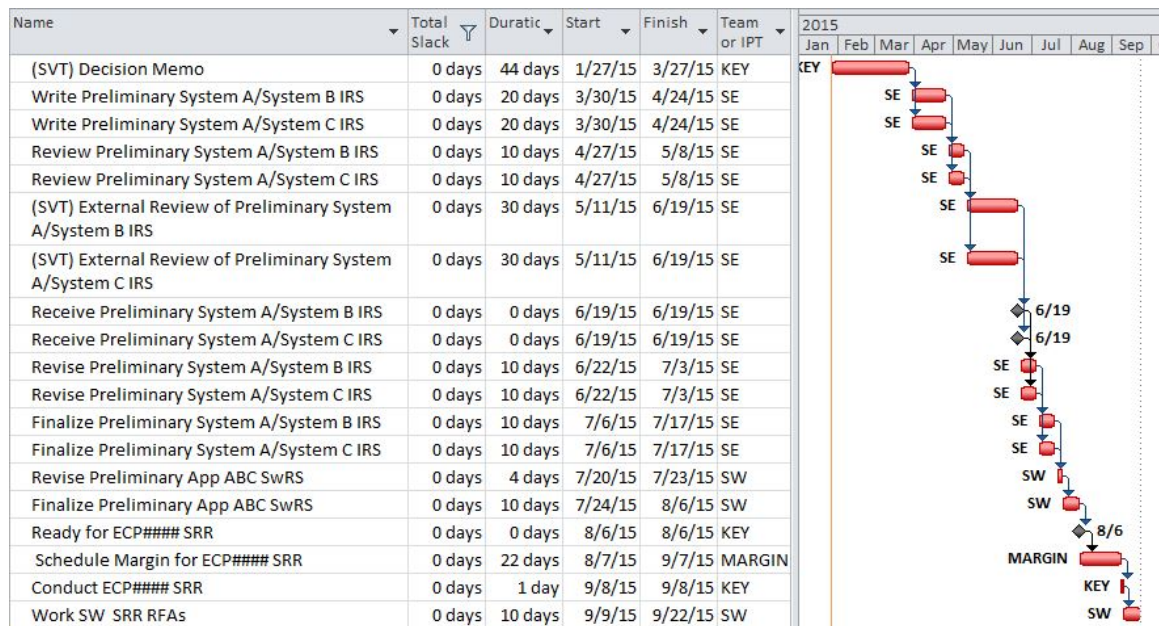
Step (d) is using the “Fill Down” command to fill all predecessor fields for Process Step 2 tasks. This automatically fills the successor field of the Process

Step 1 task which is then copied in step (e) and pasted in the predecessor field of the Process Step 3 task (“Populate database with SRR Checklist data”).

These linking operations must be repeated in various scenarios within the IMS module until all are completed. It is manual and tedious and can be performed with very little direct engineering input.

Once the manual process of linking and grouping the tasks is complete, the IMS module can begin to be used for projecting dates. Figure 14 shows the critical path to the key milestone of System Requirements Review (SRR) as an example.

Figure 14. A-BURTP Tasking Linked



A-BURTP tasking linked manually and independently by a schedule developer produces the critical path and key milestone dates.

6. Populate WBS Field with System Architecture Data

A-BURTP shall apply WBS codes where defined or system architecture information if WBS is not defined.

A-BURTP satisfies the ability to populate the WBS field with system architecture data such as the name of the component or interface, but no WBS coding was used in the simulated architecture.

7. Assign Engineering Review Type

A-BURTP shall assign review type to each task based on the review for which the task object is entry criteria.

A-BURTP met this objective by populating each Review Type field with the NAVAIR SETR event for which the task was a predecessor.

8. Populate CDRL Field with Artifact Data

A-BURTP shall populate the “CDRL” field for each artifact task to describe what object is being produced or modified by the task action.

A-BURTP met this objective by populating the CDRL field with the descriptive noun that defines the artifact.

9. Populate Resource Names

A-BURTP shall populate the “Resource Names” text field (not the “Resource Name” MS Project resource field) based on task ownership and work step information.

A-BURTP met this objective by assigning architecture to organizational area and deriving resource percentages from process steps. Resource name population is demonstrated in Table 9.

10. Populate Basis of Estimate for Original Duration

A-BURTP shall populate the “Basis of Estimate for Original Duration” text field based on architecture complexity provided by engineers.

A-BURTP met this objective by populating the Basis of Estimate for Original Duration text field in Table 9 with “System A/System C Interface complexity rated 'High' during initial interview; Complexity Factor = 2.”

11. Populate Team or IPT Field

A-BURTP shall populate the “Team or IPT” field based on architecture assignment to team or IPT.

A-BURTP met this objective by assigning architecture to organizational area and mapping task ownership to architecture ownership to populate the Team or IPT field as demonstrated in Table 9.

12. Populate Schedule Risk and Opportunity Field

A-BURTP shall populate the “Risk” text field based on risk and opportunity input from engineers.

A-BURTP met this objective by populating the Risk text field in Table 9 with “System A/System C Interface Schedule Risk/Op Band rated High/Low; RiskFactor = 2; OppFactor = -0.1.” This provides stakeholders with the rationale used by A-BURTP for calculation of optimistic and pessimistic durations.

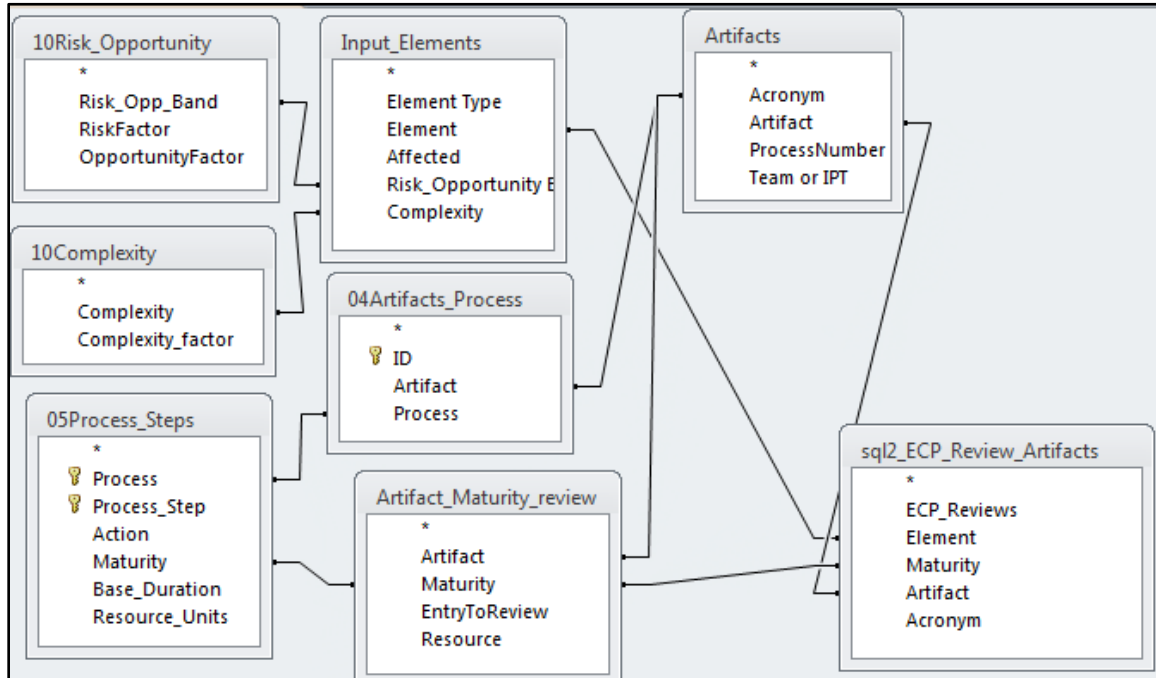
F. METHODOLOGY SUMMARY

While several other tables exist in A-BURTP, Tables 3 through 9 are the significant ones. The information engineers would normally have to consolidate for the SEP and then manually detail out to populate a NAVAIR TPW has been consolidated within A-BURTP. This data concerning architecture, artifact, local process and technical reviews now resides within A-BURTP and can be queried to produce TPWs. Figure 15 shows how these tables combine in the query that creates the artifact tasks.

When an ECP on an existing system arises, engineers determine the subset of the system architecture affected. The output of the initial architectural

assessment by the engineers will allow the scheduler to select the appropriate elements in A-BURTP and run the Artifact Task Query Set to populate the TPW. The resulting TPW contains the correct tasks associated with that subset of architecture.

Figure 15. A-BURTP Query Producing Artifact Tasks



Tables within the A-BURTP query that produces artifact tasks are shown with the field relationships.

By decomposing the Create Tasks function from Figure 8, lower-level functions were determined suitable for allocation to automated processes. A-BURTP was constructed to store the basic data the engineers use to create tasks. Common data from the SEP, along with data from engineering processes and system configuration items, was stored in A-BURTP tables. Queries were developed to perform the lower-level functions and the outputs were imported into the IMS.

Data sources and significant table construction were discussed and the query output was demonstrated to be suitable for import into the IMS. The

imported data was also demonstrated to be useful to enable the scheduler to rapidly sort, filter, group, and link the tasks to create an IMS file suitable for team review.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. CONCLUSIONS

A. RESULTS AND RECOMMENDATIONS

1. Results

The ability to determine and create the artifact tasks based on a preliminary engineering assessment of impact to the system architecture has many benefits. Uninterrupted engineering assessments, concise and unambiguous task names, maintaining SEP/IMS alignment, and one time entry for standard processes are some of the benefits.

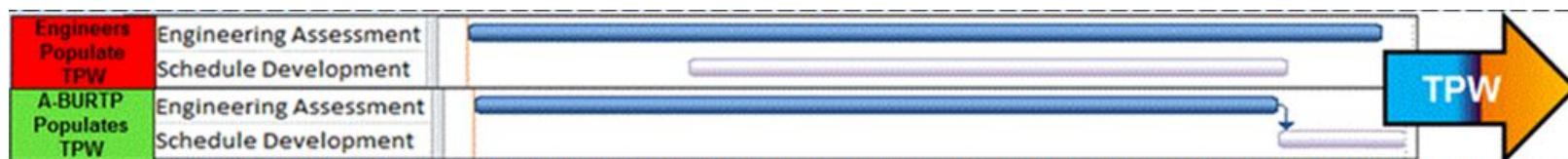
a. Uninterrupted Engineering Assessments

The A-BURTP TPW contains the same information as TPWs created using current methods. But, since A-BURTP can create the TPW so accurately and quickly, the engineering assessment of an ECP can take place uninterrupted by cost and schedule estimates.

Figure 16 illustrates notional timelines for using engineers or A-BURTP to populate TPWs. A-BURTP allows engineers to remain uninterrupted by cost and schedule data requests during their assessment of architectural changes and decisions on review tailoring. Once the preliminary engineering work is completed, A-BURTP rapidly creates the TPW. The scheduler can then import the tasks into the IMS and link, group, and sort them quickly. This new IMS tasking can be assembled within a few hours into the main IMS and ready for the engineers to review.

This rapid creation of tasks also reduces the time a scheduler spends typing and inventing task names. More time is available for schedule analysis and status updates.

Figure 16. Notional Timelines



Notional timelines show difference in engineering focus when preliminary ECP assessment is uninterrupted and schedule development is performed rapidly using A-BURTP.

b. Concise and Unambiguous Task Names

Since the task naming convention is programmed into A-BURTP, the task names are always constructed correctly and uniformly. There is no lost time thinking and discussing what to name a task. Task names are short yet descriptive enough to communicate the exact action on the exact object.

c. Schedule and Systems Engineering Plan Agreement

Maintaining alignment between the SEP, IMS, and other management tools is important to project management and systems engineering efforts. Using A-BURTP helps maintain this alignment by using the artifact listing and change management processes called out in the SEP to create the correct tasks in the IMS. Also, the SETR process entry criteria called out in the SEP comes through in task dependency attributes and ensures event-driven schedules are constructed. SEP/IMS alignment is further enhanced by the ability to rapidly update the IMS and use the latest forecast dates where required in the SEP.

d. One-time Entry for Repeating Process Steps

Since the task names are concatenated from data in several tables, process step data for mandated processes must only be entered one time. Table 6 gives two examples of repeating task strings. One of these is a 7-step process for writing and approving an interface requirements specification (IRS). Once these 7 steps were added to the Process Steps table, they are applied to each IRS affected by the ECP. The steps do not have to be looked up each time an ECP is issued.

2. Recommendations

The most significant recommendations from the research and development of A-BURTP involve process improvements to existing business practices. Integration of business areas such as design engineering, systems engineering, scheduling and cost estimating require that data can be transferred between the areas in a consistent manner. Discussions and interviews between

engineers and cost/schedule estimators may never be fully eliminated. However, developing a corporate culture that uses data transfer and business rules might help reduce the number of opinions that go into estimates and decisions.

a. *Use a Database Tool to Create Schedule Tasks*

Based on the prototype tool demonstration, it is recommended that A-BURTP be adopted by any command interested in consistent, quick IMS construction while reducing engineering time on IMS development. Programs that must continually estimate ECP impact should consider using database tools to determine task lists. Round Robin schedule reviews, engineering interviews, and TPW spreadsheets can still be used effectively to evaluate A-BURTP outputs, but cannot build initial schedules as quickly and accurately.

b. *Use a Database for SEP Sections 4.3, 4.4, 4.5, 4.6*

Systems engineers and program managers who do not have relational databases or tools for tracking their reviews, documentation, and change processes might have a difficult time populating SEP Section 4. By design, the SEP is a "living go-to technical planning document" (OSD 2011a, 6) and is intended to be data-driven (2011b). The guidance on the 2011 OSD SEP template expressly indicates that a data-driven SEP was a main intent of revising the outline (OSD 2011b).

The SEP Outline was formally released over PDUSD (AT&L) signature as an "Expected Business Practice" for immediate implementation. This streamlined SEP has been developed as one of the process streamlining initiatives under Dr. Carter's "Better Buying Power" initiative. The intent in revising the SEP is to make the document more effective, more data driven and more directly useful to programs in execution. (OSD 2011b)

A-BURTP takes advantage of the information contained in an up to date SEP and makes it more useful because it gives the scheduler a starting point.

These tables can be stored in an existing relational database or A-BURTP could be modified to contain the data and populate the SEP sections. The flow of data is important because the IMS must follow the SEP with regard to work scope. Putting the one-time effort into getting the tables right and establishing the data library can reduce time for developing cost and schedule estimates on ECPs.

If tables are created to list documentation artifacts, define standard review entry criteria, define baseline control artifacts, and define change management procedures, automating IMS task development becomes much more viable. Publishing these defined rules of engagement reduces team confusion. Section 4 of the SEP ultimately provides input to the IMS and cost estimate even as the IMS and cost estimate provide input to the SEP in other sections.

c. Review and Update Local Processes

In the process of mapping documents to local procedures and technical review entry criteria, it may be discovered that additional information is required. It is recommended that local process work steps be reviewed and updated to include clear maturity levels of artifacts, nominal durations, and resource estimates. Local processes and directives similar to those found in NAVAIR can be used by the scheduler to derive IMS tasking. Instead of finding a single document that describes the entire process, a scheduler will likely have to locate several sub processes and tie them together. Finding and connecting all the applicable guidance will be challenging. Some of the processes may not mesh and may need to be changed. These can be updated and the overall systems engineering process can be improved.

d. Export from Model-Based Systems Engineering Tools When Available

While not necessary to populate the A-BURTP tool, MBSE software can be used in many ways and having a model of the baseline system provides a place to start from when adding a function or replacing an obsolete COTS item. A system architect can examine the existing system architecture model and

determine which configuration items will be impacted to accomplish the ECP. Changes could be functional or simply a need to replace obsolete components while maintaining existing functions. In either case, a functional assessment of the change and an assessment of the required changes to the system are necessary.

Appendix A section D discusses this possibility for data reuse. Once the impact of the change has been assessed based on the system model, a list of affected architecture and artifacts could be exported from the modeling tool and required technical reviews could be determined. With the data export and the required technical reviews established, the scheduler could use A-BURTP and have the tasks in the schedule in a few hours.

Inserting the tasks in the schedule may require help from the engineers and systems engineer to determine a strategy to integrate the change into existing work, but most of the cost and scheduling work could be completed from the modeling tool export.

B. DISCUSSION OF RESEARCH QUESTIONS

1. Primary Research Question

The primary research question was, *“Can IMS tasking be derived from sources other than direct engineering input?”*

This question is answered in the affirmative. IMS tasking can be derived by combining the output products of engineering assessment of needed system architecture changes with instructions and directives. MBSE tools can also be used as part of the process as demonstrated in Appendix A.

2. Supporting Research Questions

What data is available and useful to schedule developers?

Data available includes instructions on review entry criteria and work steps in standard processes. This includes technical review directives and standard

work packages. Lists of system configuration items and the artifacts that describe them can be combined with work steps and entry criteria to derive the actions and dependencies needed for a TPW.

What other data is needed to construct IMS tasks?

To fully construct a single IMS task, the complexity, risk, and opportunity must be evaluated. Also, some form of base duration and resource requirements must be provided. IPT ownership/responsibility must also be defined.

Can IMS task creation be automated?

Can predecessor/successor information be determined?

Can task durations be estimated?

These questions are all answered in the affirmative by the work in this thesis. However, complete automation of durations and predecessor/successor logic will not always yield perfect results. The initial creation and linking of the IMS tasking can be performed very quickly and independently but some hand-offs between artifact task strings may not be readily defined in the work steps.

Can task names be standardized?

Task names can be standardized and automated and become less ambiguous in the process. The ability to construct precise task names without having to think about what to name every task is a major advantage of the A-BURTP tool over existing methodology. Some words like “draft” or “test” can be a noun or a verb or an adjective. Automating task name creation allows words like these to always appear in the exact same context (or not appear in other contexts) and become easier for the team recognize.

What other IMS task attributes can be derived from available data?

In some cases, some of the sequence of tasking should be able to be derived based on architecture. For example, if something is “built in” something else, the internal object might need to be completed first. If data is transferred

from one system to a network, the receivers need to be completed before the network can be tested.

Also, complexity of interfacing components or systems and the number of interactions between them could be factors used in developing a risk/opportunity band assessment (Tan 2012). Tan's thesis explores this concept and combining her research with MBSE and A-BURTP would be an opportunity for future work.

C. BENEFITS OF RESEARCH

1. Reduce Non-Value-Added Activity

Allowing engineers to work on complex problems is of much higher value than having them type tasks in a spreadsheet or schedule. Use of A-BURTP puts the architectural assessment first, avoids repeated inputs, omissions and mistakes by leveraging off of one-time process step inputs and allows the engineers to focus uninterrupted on engineering issues.

The methodology employed in this thesis not only reduces engineering NVA, but also reduces schedule developer NVA. Reducing the time to think of task names and type them in a project schedule is beneficial as is the availability of consistent linking information. Schedule developers also need time to analyze the IMS and ensure schedule health.

The ability to rapidly create schedule tasks traceable to system architecture and organizational business rules allows the schedule developer to integrate the work into the program resources and other constraints. Rather than slowing the team down to create tasks, the schedule developer becomes an asset to the team by providing a product to review and adjust to capability delivery needs.

2. Rapid Response and Enhanced Agility

There is great agility in being able to rapidly construct an IMS module without disrupting engineers. Electronic systems must change rapidly due to both

technology advances and threats. Automation of efforts that follow defined rules is a way to accelerate the processes that update these systems.

3. Integration of Management Tools

Having all the management tools in agreement is a way to help the PM and SE stay in agreement. Strong, consistent leadership is needed for integrated product teams consisting of members with strong allegiances to technical disciplines. In NAVAIR, there is great benefit of tying together NAVAIRINST 4355.19 and local processes to produce a data-driven SEP that follows the OSD template. Not only can schedule developers derive tasking, but new team members can read and search the SEP and ramp up quickly.

D. TOPICS FOR FUTURE RESEARCH

1. Improvements to A-BURTP

The development of A-BURTP for this thesis provided a useful tool to automate task planning worksheets. However, there are many improvements that could be made to make A-BURTP even more useful.

a. Include Test Events

Future researchers could add Test event task creation as a new function of A-BURTP. “Verified by” reports from MBSE could be analyzed for the necessity of test events. Review of organizational directives for test processes, plans, and reports would need to be conducted in order to capture and store the actions required.

b. Evaluate Complexity and Risk

Use MBSE model output to derive complexity of ECPs based on aggregate changes to architecture, associated interfaces, complexity of data packets, and SOS interconnectivity.

In NPS thesis *Application of an Entropic Approach to Assessing Systems Integration* Tan presented an entropic approach to evaluating risk to successful

integration based on interface complexity. Tan demonstrated that as complexity increases, the probability of successful integration decreases. Tan's recommendations for future work included usage of real data.

Tan's approach is appealing as method to determine IMS schedule margin durations to account for schedule risk. Tan has a forward-looking approach that could be applied if the IMS/IGS nodes and varying degrees of interaction between them could be quantified (Tan 2012).

One approach would be to follow Tan's method using the MBSE tool to simulate the risk and determine three-point duration estimates (Tan 2012). The results could be used to run SRA and determine high risk areas to incorporate schedule margin or prioritize interface planning and agreements.

This information could be combined with higher-level deadlines to determine the latest possible date for each MOA signature milestone. Of even more value is the ability to automatically populate the "Impact if Not Completed" field in the OSD MOA table. This would fulfil the requirement for OSD SEP template table 2.1-1 Required Memoranda of Agreement of the OSD SEP Template shown in Figure 17 (OSD 2011a). The ability to state the risk and impact of a late interface MOA based on system architectural data provides confidence in briefing needs for help to higher echelons.

Figure 17. Schedule Risk Assessment (Adapted from OSD 2011a)

REQUIRED MEMORANDA OF AGREEMENT				
Interface	Cooperating Agency	Interface Control Authority	Required By Date	Impact if Not Completed

Table 2.1-1 Required Memoranda of Agreement (mandated) (sample)

Expectations: Programs whose system has external interfaces need to have dependencies (i.e., hierarchy) clearly defined. This should include interface control specifications, which should be confirmed early on and placed under strict configuration control. Compatibility with other interfacing systems and common architectures should be maintained throughout the development/design process.

MBSE complexity and risk data could flow through IMS to enable schedule risk assessment and prioritization of interface planning to populate OSD SEP Template MOA table 2.1-1.

c. Improve the Tool Design

A-BURTP was built based on research of systems engineering rules and scheduling best practices rather than a disciplined programming approach. An attempt at data normalization was made, but a fresh start with knowledgeable computer programmers or architects could create a better product.

User interface is directly in the tables and there is no user-friendly way to operate A-BURTP. There are most certainly better ways to construct queries. For NAVAIR, there may be a better way to construct the SETR logic and formulate how the Review Type field is populated. The present queries assign the planned reviews to the Review Type field. A better way might be to create “Ready For” milestones for all SETRs to allow visibility into the process.

For example, the draft software test plan (STP) is entry criteria to System Software Review (SSR) and the final is due at Critical Design Review (CDR). If the SSR is not conducted, the draft STP is due at Preliminary Design Review (PDR). If a combined PDR-CDR is conducted, the STP does not gate any reviews until CDR. Having a “Ready for SSR” milestone in the path might be of value even if the SSR is not conducted.

2. Review NAVAIRINST 4355.19e and Explore the Ability to Incorporate the Release Backlog Review (RBR)

The 2015 release of NAVAIRINST 4355.19e includes the RBR for using the Agile software development process (NAVAIR 2015). This thesis was written with NAVAIRINST 4355.19d and was in review when the NAVAIRINST 4355.19e was released. The 2015 release may be even more easily converted to populate database tables and produce IMS tasking. The RBR event was not explored in writing this thesis and may require special handling to incorporate.

Agile development can be applied to other disciplines besides software (Alberts 2011). There is a need to understand the RBR and how it behaves with regard to cost, schedule, and performance.

3. Smash Bureaucracy

Garry Newton, Deputy Commander of NAVAIR, sent a “Smashing Bureaucracy Update” on February 20, 2014, in which he stated:

“Some of our processes are in place for good reason, but others have evolved into self-imposed checklists that go beyond the original intent or are no longer necessary. We’re looking for ideas with a significant potential return on investment, as well as smaller changes that could impact a large percentage of the workforce.” (Newton 2014)

One finding during the testing of A-BURTP was the number of tasks produced for a single ECP. It takes 347 tasks per ECP before adding any architecture and artifacts. These tasks include administration tasks for ECP configuration control board (CCB) meetings and just four SETRs of SRR, PDR, CDR, and TRR. Tasks include checklists, meeting minutes, slides, dry runs, slide updates, actual events, and RFAs.

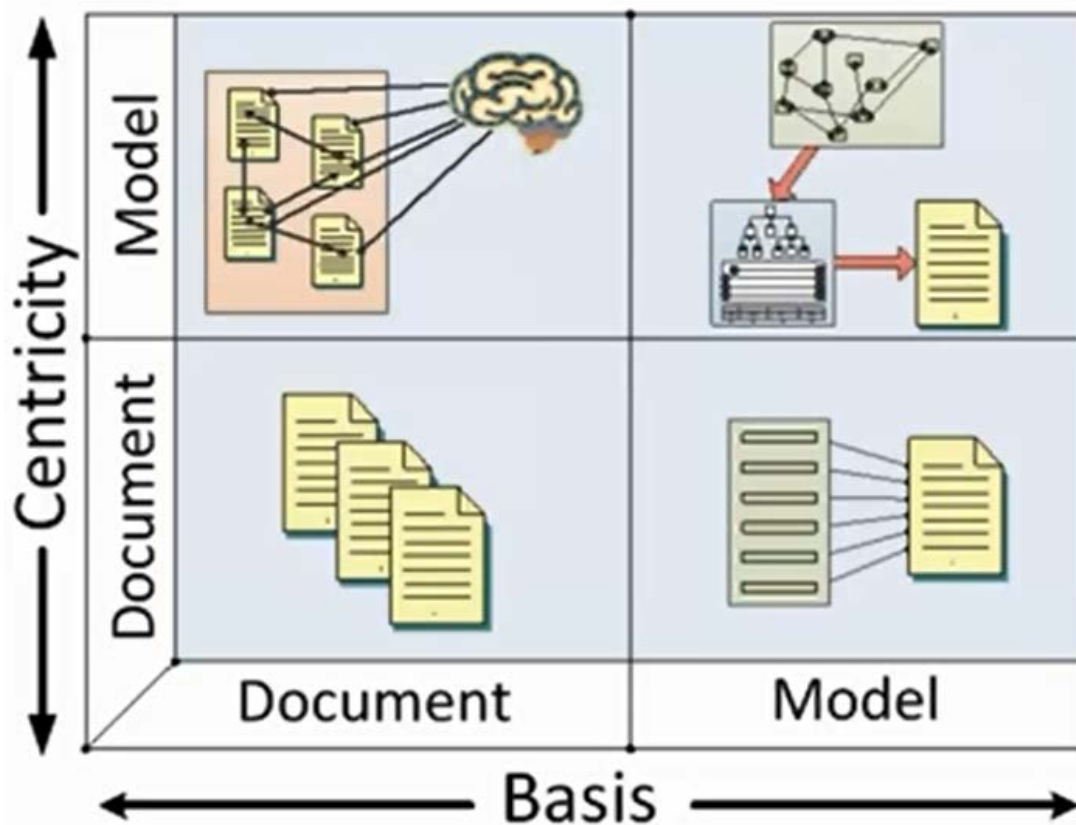
A potential return on investment for the effort to construct NAVAIR architectures in modeling tools is the possibility to move away from slide-based technical reviews and move to model-based technical reviews (Vitech 2012).

Reviews of the MBSE file can be conducted at any time and live reviews by technical boards can field questions and identify gaps in real time.

A model-based file of a SOS member system can be shared with other members to enhance understanding and compare interface compliance. Higher-level integration reviews can collect models from lower-level systems and evaluate compliance.

Figure 18 is from Vitech presentation “Document the Model, Don’t Model the Document” that discusses the systems engineering basis and centrality. The ability for systems engineers and architects to present a live model rather than a slide deck could replace some of the “death by PowerPoint” events with interactive meetings actually looking at system architecture and traceability. The “real” work still needs to be done and using engineers to design systems rather than create slides would be a move in the right direction.

Figure 18. “Document the Model, Don’t Model the Document”
(Source: Vitech 2012)



Vitech Corporation's “Document the Model, Don’t Model the Document” presentation discusses basis and centricity of systems engineering methods and encourages a model-based, model-centric method.

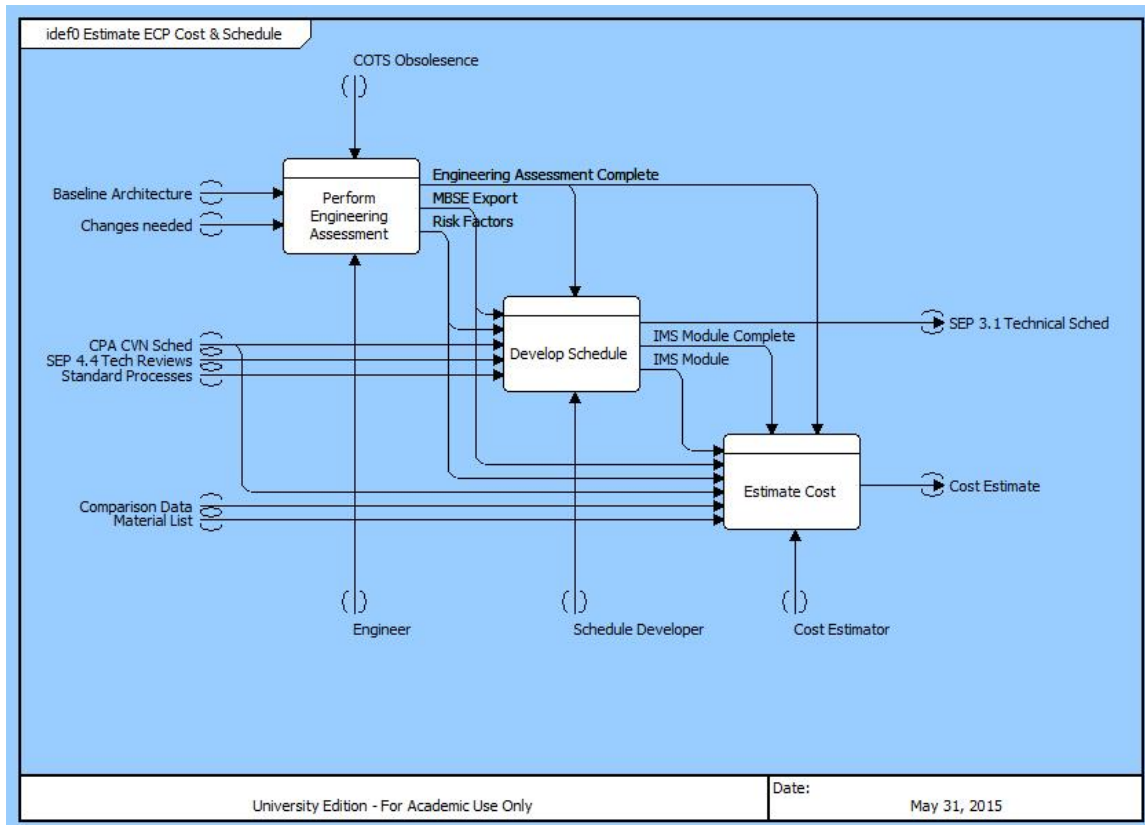
4. Cost Estimation

Figure 19 illustrates a proposed methodology and process flow for performing ECP cost estimates using MBSE tools and A-BURTP. After a system model was created in a modeling tool, the data could be exported and provided to the schedule developer. The model data could be imported into A-BURTP by the schedule developer and a schedule module for the IMS could be developed. Cost estimators can then use both the MBSE data and the schedule resources as subsets of the data needed to produce a cost estimate.

If this concept was used in conjunction with the recommendation of modeling CVN SOS architecture in MBSE, the impact to each ship and the

number of ships affected by the change could be assessed quickly with data-driven queries in A-BURTP. Cost estimators could use the architecture data for estimating procurement of test and lab assets as well as for estimating equipment quantities for ship installation events and spare parts inventory.

Figure 19. Proposed Data Flow



The proposed data flow from engineering evaluation to schedule development and cost estimation shows the potential for rapid assessment of impact to cost and schedule for ECPs.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. MODEL-BASED SYSTEMS ENGINEERING EXPORTS USED IN RAPID SCHEDULE DEVELOPMENT

A. MODEL-BASED SYSTEMS ENGINEERING (MBSE)

MBSE tools are relational databases used to manage requirements and model relationships between elements associated with system architecture. These elements include, but are not limited to, requirements, functions, components, documents, interfaces, links, inputs, outputs, triggers, and resources. Very specific modeling language is used to relate each element to one or many other elements within the system and its environment. Vitech CORE9 is a MBSE tool and the academic license version was used to model CVN IT system architecture and export the sample data to A-BURTP. This appendix explains what MBSE data was exported, how it is used in A-BURTP, and how the data was created.

Table 14 was exported using a standard export format available in the CORE9 university license. Only three of many data elements available in the MBSE file were exported. First, the “Name” field contains the unique names of artifacts associated with the system. Next, the “classified by” field contains the type of each artifact. Finally, the “Documents” field contains the system architecture element documented by the artifact. The first row in Table 14 is interpreted as, “The A.1.2 Cabling is documented by the Block Wiring Diagram drawing.” Each row follows the same format.

Table 14. MBSE Architecture Elements Assigned “Documented by” Relationships

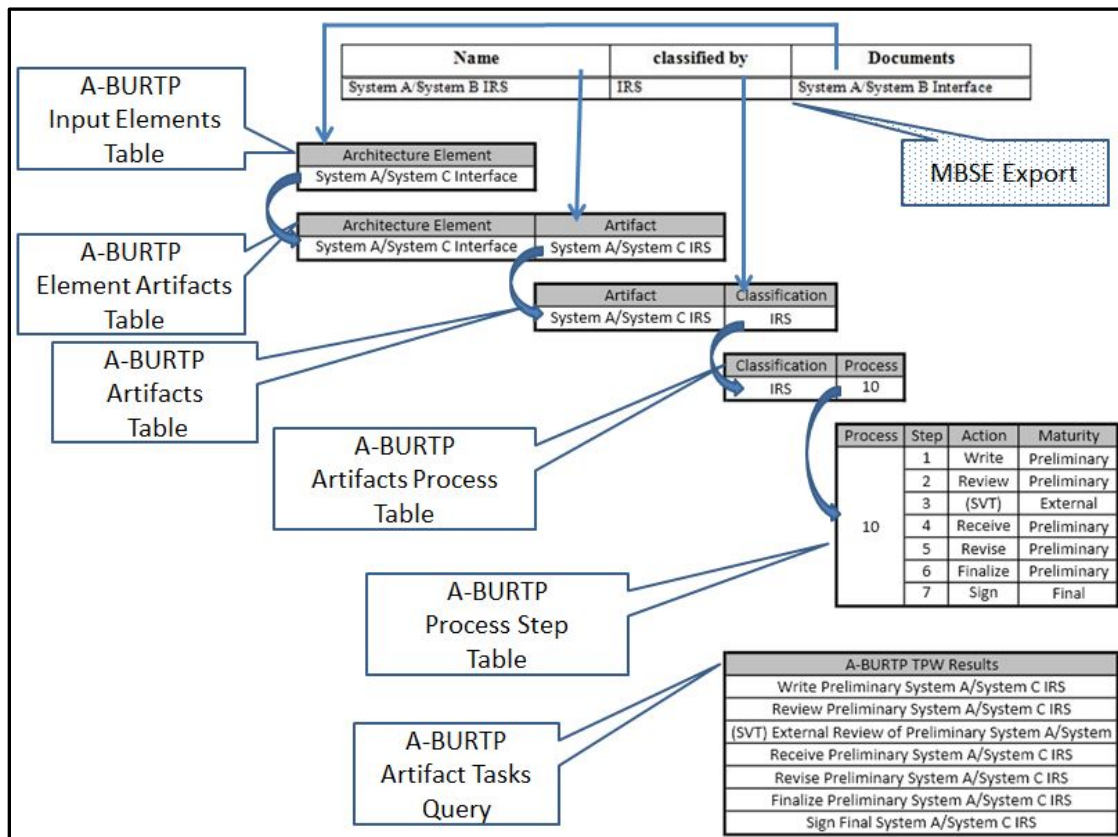
Name	classified by	Documents
Block Wiring Diagram	Drawing	A.1.2 Cabling
Cooling Fan Component Spec	Component Spec	A.1.1.2 Cooling Fan
CSCI 1 SwRS	SwRS	A.2.1 Operating System
CSCI 2 SwRS	SwRS	A.2.2.1 Application ABC
CSCI 3 SwRS	SwRS	A.2.2.2 Application XYZ
Network Switch Component Spec	Component Spec	A.1.1.4 Network Switch
Rack Assembly Drawing	Drawing	A.1.1 Rack Assembly
Server Specification	Component Spec	A.1.1.1 Classified Server
System A/CVN IRS	IRS	System A/CVN Interface
System A/System B IRS	IRS	System A/System B Interface
System A/System C IRS	IRS	System A/System C Interface
System Requirements Specification	SRS	A.0 System A
UPS Component Specification	Component Spec	A.1.1.3 UPS

MBSE architecture elements that have been assigned “Documented by” relationships are exported for use in IMS development.

Table 14 represents a master listing of all artifacts documenting the fictional system developed in this appendix. It can be seen that some of the architecture elements in the Name field are system components while interfaces, assemblies, and the system itself are also included in the export. These architecture elements are the reason for the “Architecture-Based” title A-BURTP.

Figure 20 illustrates how the MBSE export is distributed to the A-BURTP tables. Also shown is how this data is mapped through the processes and process steps within A-BURTP to determine the correct IMS tasks. The resulting TPW contains the correct tasks associated with that subset of system architecture.

Figure 20. Mapped MBSE Architecture and Artifact Data



MBSE architecture and artifact data is mapped to process data within A-BURTP to produce IMS tasks.

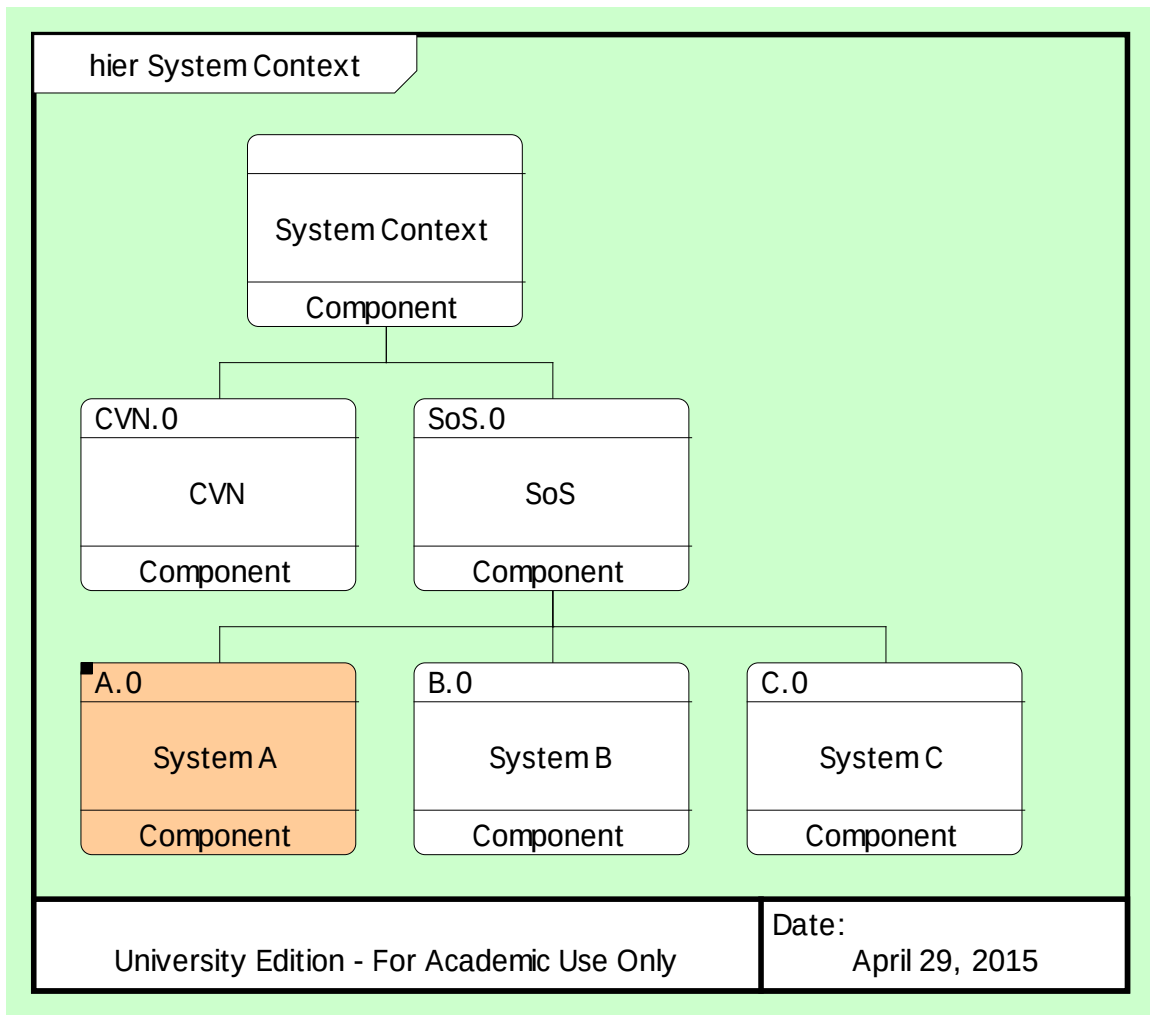
While the master list allows A-BURTP to create every possible task that could be performed on every system artifact, the user of A-BURTP reduces that list by selecting the subset of architecture affected by an ECP. In a real system with a real MBSE file, a much larger list would be exported to initially populate the Input Elements table and the Elements Artifacts table. When engineers evaluate a proposed ECP for its impact on an existing system, the subset list of affected elements would allow the scheduler to use A-BURTP to create ECP tasks based on that subset. This could include changes to existing elements, removal of elements, or addition of elements. New elements would require new elements to be added to Input Elements and Elements Artifacts tables.

B. CREATING A SAMPLE SYSTEM ARCHITECTURE

The correct and complete development of a MBSE file for a system goes far beyond what was done for this thesis. A disciplined and rigorous systems-engineering approach was not followed to construct this small MBSE file used for IMS development. In actual practice, a systems engineering process would be followed to define and decompose requirements and functions, allocate functions to system components, and develop defining artifacts. The simplicity of the MBSE export data makes it very apparent that this engineering rigor could go on uninterrupted by requests for cost and schedule information if an MBSE export process were used to develop IMS tasking.

Because this appendix focuses on the ability to export architecture and artifact data to use for IMS tasking, only specific areas of the MBSE file applicable to that end were populated. A “middle-out” approach (Vitech 2013c, 1) was used to construct a limited model of a SOS member-system architecture within a SOS. Figure 21 highlights the fictional System A within the SOS context.

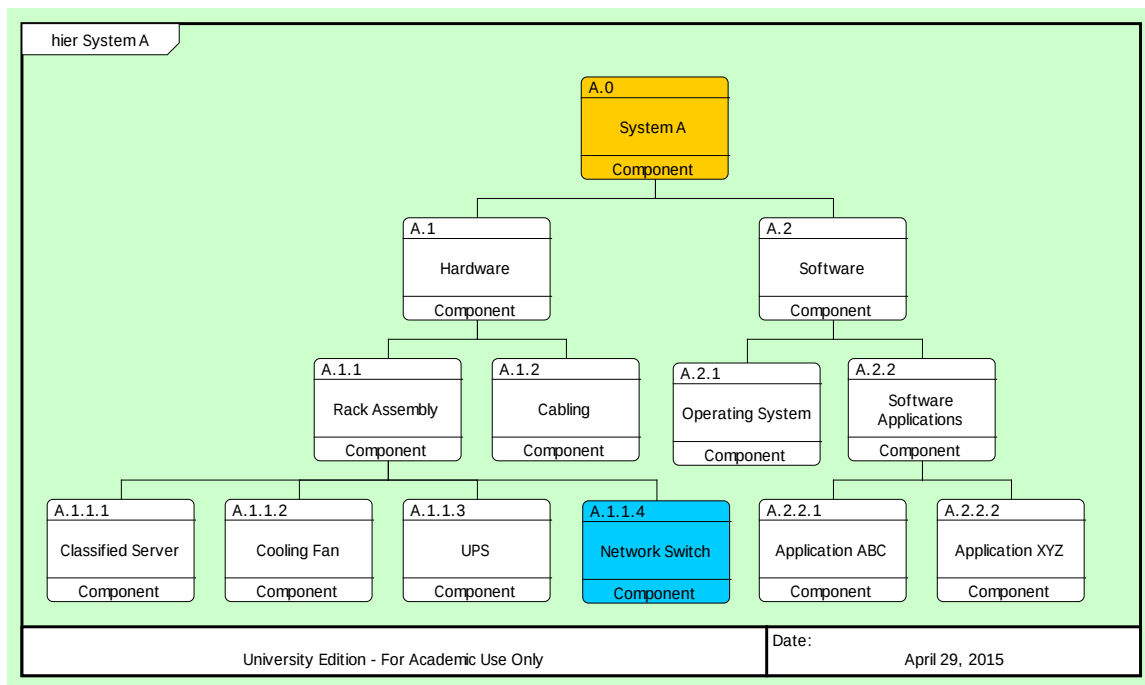
Figure 21. System Context for “System A”



The system context for “System A” is modeled in the Vitech CORE9 MBSE software.

In Figure 21, the black square in the upper left corner of the System A node indicates further decomposition exists in the MBSE file. Vitech CORE9 uses the “built in/built from” relationship to decompose higher level components into lower-level components. Figure 22 shows the lower-level decomposition of System A including hardware and software components.

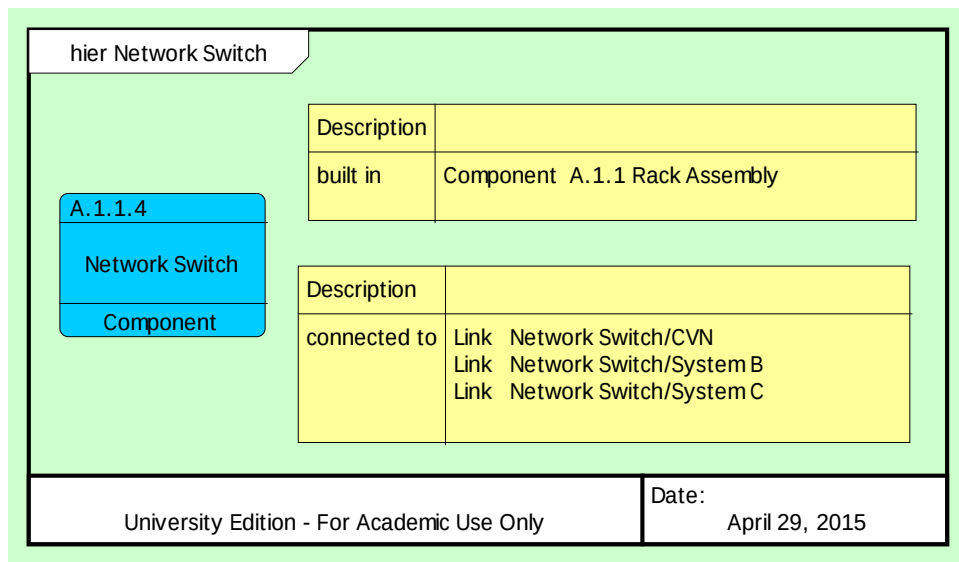
Figure 22. “System A” System Architecture Hierarchy



“System A” system architecture hierarchy shows the lower-level configuration items created in the Vitech CORE9 MBSE tool and highlights the network switch.

System A has interfaces with other fictional systems. These were created in Vitech CORE9 using the “connected to” relationship. Figure 22 highlights the network switch within System A. Figure 23 shows the network switch “built in” the System A rack assembly and “connected to” links with the CVN, System B, and System C.

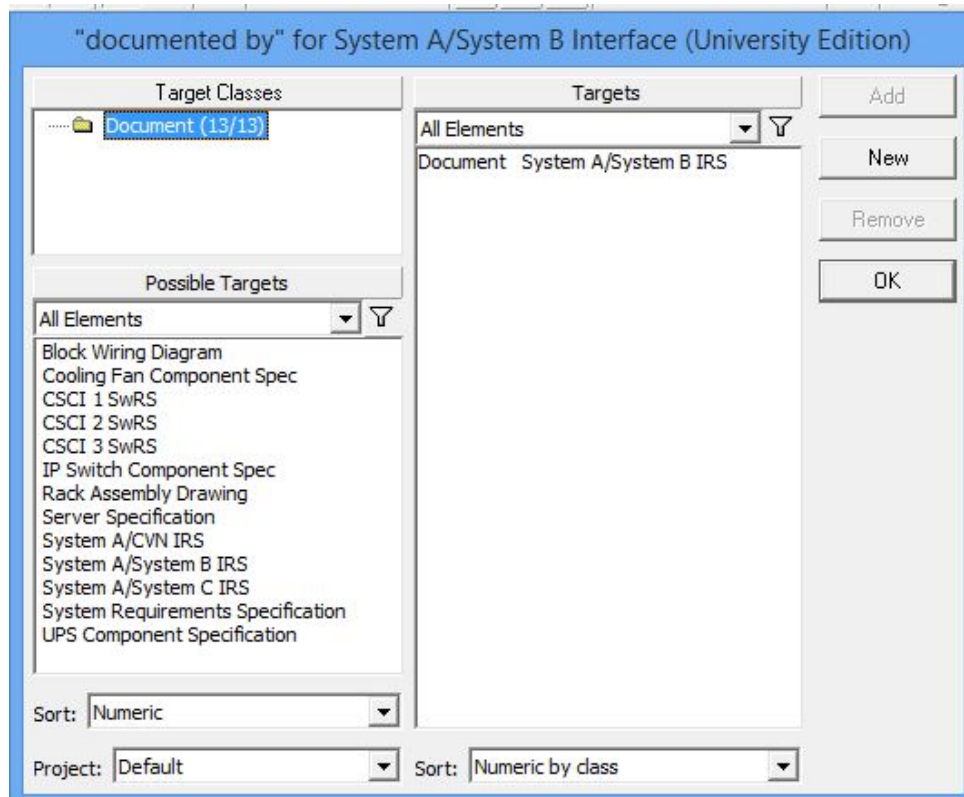
Figure 23. Network Switch



The network switch is built in the System A rack assembly and connects to the CVN, System B, and System C.

With the System A architectural components and interfaces defined, the next task is to identify the artifacts associated with System A. Figure 24 is a screen capture from Vitech CORE9 where the tool is being used to assign the “documented by” relationship. In this case, the System A/System B Interface is documented by the System A/System B interface requirements specification (IRS).

Figure 24. Relationships between Architecture and Artifacts



The relationships between architecture and artifacts are assigned using the “documented by” screen in Vitech CORE9.

This “documented by” relationship, and its inverse: the “documents” relationship, are used in Vitech CORE9 to establish traceability between system architecture and governing documents. While Figure 24 only shows the System A/System B IRS document assigned to the System A/System B Interface, the remaining documents carried in the MBSE file can be seen in the left hand panel. These were also assigned to applicable system components and elements using the same process.

Assigning the “documented by” relationship is the end of input for the Vitech CORE9 model for this appendix. Even at this interim stage of MBSE development, much can be derived about the work to be captured in the IMS based on just the architecture and artifacts. The next step exports the MBSE data that is significant to IMS development.

C. LITERATURE FOR APPENDIX A

1. DOD Architectural Framework 2.02

Version 2.02 of the DODAF is an immense resource describing various required viewpoints from which DOD systems must be assessed. The DODAF conformance is mandated by the DOD Chief Information Officer and described as:

DOD Components are expected to conform to DODAF to the maximum extent possible in development of architectures within the Department. Conformance ensures that reuse of information, architecture artifacts, models, and viewpoints can be shared with common understanding. Conformance is expected in both the classified and unclassified communities, and further guidance will be forthcoming on specific processes and procedures for the classified architecture development efforts in the Department. (DOD 2015)

Relevance to Thesis:

The OSD SEP Section 2.1 requires programs to address efforts towards compliance with the DODAF (OSD 2011a). MBSE tools are designed to produce reports conforming to DODAF standards (Vitech 2012). Constructing system architecture in MBSE tools is a direct engineering effort which can be reused to develop IMS tasking for ECPs.

2. NASA WBS Handbook

The well-known relationship between system architecture and schedule development is demonstrated in the use of work breakdown structure (WBS). Section 4.3 of NASA Special Publication WBS Handbook describes the WBS and schedule interactions as:

The WBS provides a framework for detailed project planning and schedule development. As WBS elements are established and work content is clearly defined in the WBS Dictionary, it is then possible for the project team to determine the tasks (activities) and events (milestones) that are required to successfully complete the project goals and products. Tasks and events are identified for the effort contained in each lowest-level WBS element. ...Since a

product-oriented WBS serves as the framework for schedule development, then resulting project schedules are also product-oriented. (NASA 2010)

Two main points are applied to use of MBSE in this thesis:

- 1) MBSE architecture data (exports of products required) could be reused to develop the WBS for IMS architecture and IMS tasks.
- 2) The WBS elements can be incorporated as objects into the action/object task name structure for IMS tasks.

3. Vitech CORE9 Architecture Definition Guide

The *Vitech CORE9 Architecture Definition Guide* (Vitech 2013a) provides guidance on how to use Vitech's MBSE tool to comply with DODAF version 2.02 objectives (DOD 2008). It also describes the purpose of architectures "achieving a well-defined system...for a specific timeframe..." (1).

This resource was also helpful in understanding that the complete DODAF architecture population was not necessary to provide a basic system architecture that could be used for exports into the prototype tool.

4. Vitech CORE9 System Definition Guide

The *Vitech CORE9 System Definition Guide* (Vitech 2013b) provides guidance on how to use Vitech's MBSE tool to construct system architectures.

This guide was used to facilitate creation of sample MBSE files in this thesis. Understanding the "documented by" relationship (7) was necessary to export the proper data from the MBSE tool to use in IMS task creation in the demonstration. Several specific files were constructed in the research process. Understanding of the relationship between system architecture and schedule architecture enabled automated subtask and summary task naming.

5. Using Innoslate for Program Management

Using Innoslate for Program Management discussed the capability SPEC had programmed into their MBSE tool to allow it to construct schedules for program management (SPEC 2013).

Review of this resource showed that the thought of combining MBSE with IMS development had been attempted by this company as well. Vitech CORE9 offers similar capability (Vitech 2013a), but Vitech CORE9 and Innoslate MBSE software is very complex, expensive, and not available on Navy and Marine Corps common machines.

In an organization with no constraints on MBSE/SEP/IMS software selection, use of the program management capability of these MBSE tools could be the better option. In Navy circumstances where MS Project is easily obtained and widely understood, a utility to transform MBSE data into useful MS Project data would be a better fit. However, an explanation of tool use and interface between engineering tools is necessary in OSD SEP Section 4.7 (OSD 2011a).

D. MBSE TO IMS TO SEP DATA REUSE

In this context where engineers are called on to quickly assess impacts and risks and make recommendations to the PM for execution, the PM is also required to quickly evaluate those recommendations and make decisions that allow the team to move forward. The SEP and IMS must communicate the same strategy to all concerned. Both the CPA CVN schedule constraints and the industry decisions to discontinue hardware and software are outside the control of the program. But, reducing the NVA burden of IMS and SEP development is an area of waste that can be reduced.

MBSE tools allow high-value system architecture models to be constructed and reused for other purposes thus providing additional value for the time spent on model construction. As CVN IT programs are impacted by ECPs, different areas of system architecture and artifacts require changes. Each ECP may be unique, but work steps in one SWP may be the same for many artifacts.

It is feasible that IMS tasking can be derived from the architecture, artifacts, SWP work steps, and SETR plan. Task durations could be estimated based on complexity of changes to architectural elements.

If MBSE exports were used in deriving IMS tasking in an automated or semi-automated way, both engineers and schedule developers could be focused on activities within their skillsets and therefore, be of higher value to the program. Further, if the NAVAIRINST 4355.19 SETR entry criteria were captured in the SEP and required artifacts were matured through SWPs, a significant amount of repeating IMS task creation and linking can be derived in an automated or semi-automated way without direct engineering input. If these critical interactions between MBSE, the SEP, and SWPs are defined properly, automated TPWs can be produced for import into the IMS.

An ideal management system would allocate functions to appropriate team components and allow each one to work efficiently and undistracted. While there will always be a few distractions, a well-defined and semi-automated MBSE/SWP/IMS/SEP data flow would certainly allocate more of the IMS development work away from the engineers. The IMS and SEP could then become more of a help and less of a resource drain to the engineering team.

E. APPENDIX A SUMMARY

The academic license which was used has less capability than the commercially available software but these limitations were not prohibitive to this thesis. Other MBSE products are available with similar capabilities and the concepts in thesis are applicable and transferrable to other tools.

MBSE licensing and learning curve can be expensive but the ability to export text tables makes the MBSE data useful far beyond the few MBSE experts a program may provide with software licenses. MBSE tools include requirements management capability and enable the creation of flow diagrams, documents, drawings, and specifications

One obvious means to reduce engineering effort expended on IMS and SEP maintenance is to pull information needed for SEP and IMS development from engineering sources other than the engineers themselves. One of those sources is MBSE exports such as the data contained in Table 14. MBSE export is the input generating mechanism used in the demonstration example provided in this thesis. Alternative mechanisms which generate the same tool inputs could also be used.

MBSE exports are a natural fit for generating IMS input because MBSE software tools require the application of engineering rigor to step through the systems engineering process. This results in well-defined system functions that are based on requirements and allocated to components which, in turn, are defined by artifacts such as drawings and documents. As a system's MBSE file begins to mature, exports become available for other uses. Some of the work performed by the engineering resources to build the MBSE model can then be reused to develop the IMS, cost estimates, or for other purposes. The ability to export is provided by several common MBSE tools.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. NAVAIR CVN SOS CONTEXT

A. NAVAIR ELECTRONIC SYSTEMS ON CVNS

The Naval Air System Command (NAVAIR) sustains multiple SOS member systems onboard the aircraft carrier fleet. With COTS obsolescence impacting all SOS members, ECPs must be evaluated continually. ECPs are also initiated due to new functional requirements, aircraft types, and cyber security impacts. The ability of SOS member systems to share MBSE files and evaluate ECP impacts can aid decisions on simulator development and testing. The shipboard SOS scenario is an ideal application of MBSE tools because an architectural model can be developed for each unique “as-is” configuration and then modified to the “to-be” configuration.

This specific context faces many challenges in creating and maintaining synchronization between the IMS and SEP. The systems face continual obsolescence of commercial off-the-shelf (COTS) hardware and software while fielding opportunities are limited to allotted periods in the Carrier Planning Activity (CPA) CVN (maintenance) availability schedule. There are only ten active CVNs and at least six must be able to be deployed within 30 days and a seventh must be deployable within 90 days (Yardley et al. 2008). These constraints greatly slow down the ability to perform major upgrades on the CVN ships and confound the ability to move to a normal Full Rate Production (FRP) phase.

These embedded systems often function together as a System of Systems (SOS) and their many program offices must coordinate changes in order to maintain interoperability. ECPs to member systems can be the result of commercial off the shelf (COTS) obsolescence or as a result of changes to functional or regulatory requirements driven from within the embedded system or by any of the interfacing systems.

The fielding of a system that will perform as a part of a SOS onboard a CVN is subject to many outside constraints. It can be difficult, if not impossible, to

bring the entire fleet to single configuration because each program's fielding schedule of new systems or updates/upgrades to all CVNs is subject to the incremental availability of each ship for maintenance and upgrades (Yardley et al. 2008).

The program SEP and IMS are updated to include the SETR events for each ECP (OSD 2011a). From a programmatic standpoint, the IMS has many starting points from the ECPs and many end points from the CVNs. Additionally, earlier ECPs must be adequately matured before merging with later ECPs. In this scenario, a program IMS is a component of a SOS IMS and a CVN IMS. The focus of this thesis is to create the bottom-level tasks in a single ECP within a program IMS. This low-level of tasking must be inserted into the program IMS in order to be integrated with the SOS IMS and CVN IMS.

B. CVN AVAILABILITY FOR MAINTENANCE

CVNs have operational availability requirements that compete with their availability for maintenance and upgrade. Specific periods are planned into the 50-year service life to sustain the CVN. The global capability need of forward presence must be balanced with the ability to maintain the fleet (Yardley et al. 2008). Required capability for the operational periods must be coordinated well in advance for systems to be prepared during the availability windows (Singh 2006). Like any maintenance schedule, these maintenance availabilities are subject to change caused by factors which can range from internal job overruns to complex international events.

There are times when formerly-immovable start and end dates on the CPA schedule are adjusted and new dates are fed back into every onboard program's IMS. This reality is illustrated in Figure 25. CVN-69 and CVN-75 swapped deployments due to overruns in the CVN-69 maintenance visit which drove changes into programs that had maintenance or modification planned for either or both ships (LaGrone 2014). Changes such as these can ripple through many programs and drive changes to respective IMS files and SEP documents.

Figure 25. IMS and SEP Review and Changes (Source: LaGrone 2014)



IMS and SEP review and changes for onboard systems are required after the decision to swap deployments between the CVN-69 and CVN-75.

C. COTS OBSOLESCENCE

While CVN deadlines impose schedule constraints, competitive forces in the IT industry drive innovation and new technology releases much faster than the CVN fleet can be upgraded. This can result in ECPs due to COTS obsolescence being worked concurrently with system development. The relatively short service lives of COTS hardware and software items present supportability and maintainability challenges when embedded on a CVN with a 50-year service life (Singh 2006).

Pameet Singh and Peter Sandborn addressed this obsolescence issue of electronic components embedded in long-life higher assemblies by developing the pro-active Mitigation of Obsolescence Cost Analysis (MOCA) methodology.

They note that it is not uncommon for embedded systems onboard ships to consist 70% of obsolete components at the time of ship delivery (Singh 2006).

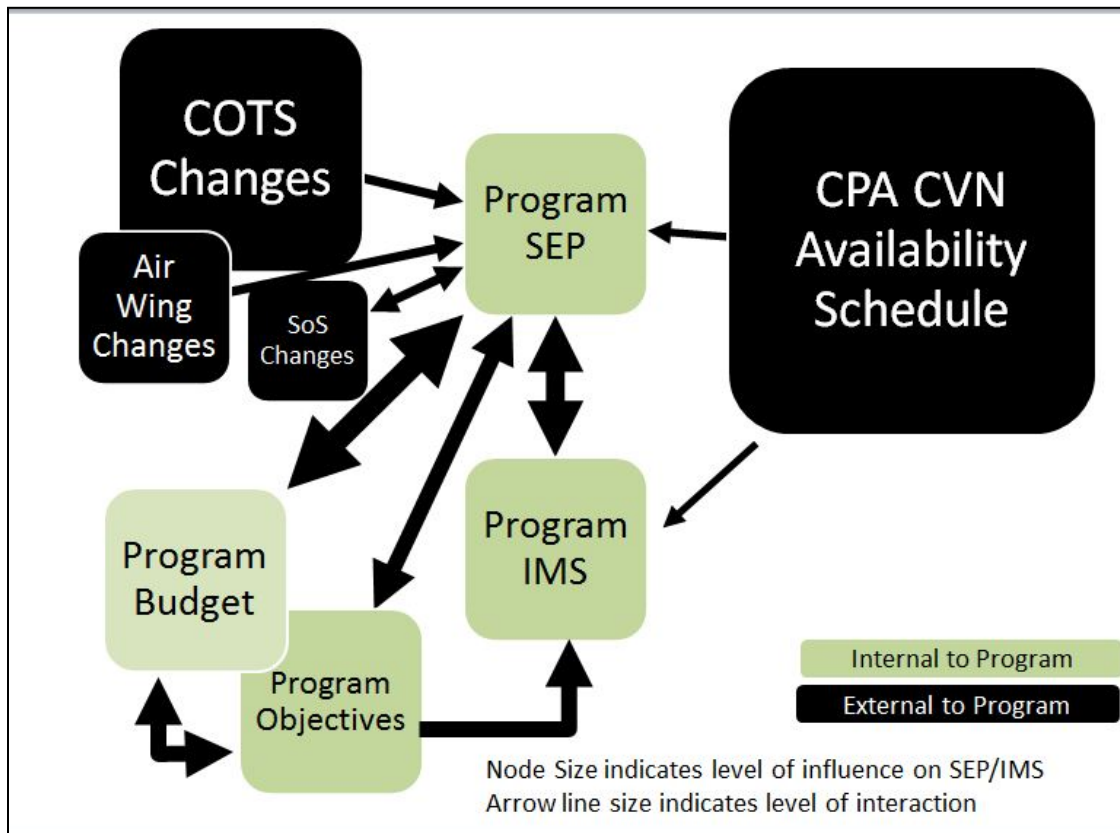
Since COTS obsolescence and CVN availability are common issues to all CVN IT systems, it is not unusual that several systems within the SOS are impacted by ECPs at the same time. SOS component systems must coordinate fielding of technology refreshes and evolving technology to ensure interoperability and supportability during the CVN at-sea period (OSD 2011a).

Design work on a SOS component system is impacted by a variety of sources internal and external to that system. Frequent updates to the SEP, IMS, and other planning documentation must be completed as a result of unplanned changes. When new requirements and COTS obsolescence-driven changes arise, each must be evaluated for the team's ability to fully develop, test, and field the resultant new system configurations within the timeframe allotted by the CVN availability schedule, the program budget, and the IPT resources. Program plans must be communicated to other programs within the SOS to coordinate capability delivery (OSD 2011a).

D. PROGRAM CONTEXT—IRRESISTIBLE FORCES AND IMMOVABLE OBJECTS

Figure 26 shows the SEP and IMS as interconnected communication tools within the context of factors internal and external to the program with varying degrees of influence and interaction. The layout illustrates the external forces of System of Systems (SOS), Carrier Air Wing (CVW), and continual COTS obsolescence driving changes to the SEP and IMS from one side while the CVN Availability Schedule imposes completion deadlines from the other.

Figure 26. Interactions between Internal and External Elements



The context diagram shows how a CVN IT program SEP and IMS interact with COTS obsolescence, the CPA availability schedule, and other internal and external elements.

The “COTS Changes” and “Air Wing Changes” nodes represent large influences on the program in the form of additional work and associated duration. COTS obsolescence is a virtually irresistible force to the program because the small production lots used by the CVN fleet often provide little to no leverage for extended supply or support negotiations with industry. Obsolescence ECPs can add work scope that must be integrated into program objectives without causing schedule delays.

Similarly, the “CPA CVN Availability Schedule” is represented as a large node with high influence and one-way push on the program. While the CPA does adjust the CVN availability schedule, embedded system programs must adjust to those moves. Any IPT that does not want their program to be the cause of such adjustment must consider CVN availabilities to be immovable objects.

“SOS Changes” shows a two-way interaction because SOS member programs can negotiate and coordinate amongst themselves to some degree. It should be noted that the impact and interaction of COTS changes and CPA deadlines to the SOS and Air Wing are assumed to be captured internally to those nodes and are intentionally excluded from Figure 5.

E. PMA-251 CONFIGURATION MANAGEMENT PLAN (CMP) FOR LAUNCH AND RECOVERY EQUIPMENT

The PMA-251 CMP was reviewed to determine low-level process steps for ECPs which were incorporated into the tool process data and used to admin tasks. PMA-251 manages Aircraft Launch and Recovery Equipment (ALRE) and contains several information systems which operate together in a SOS context onboard CVNs. This CMP describes the PMA’s local application of the process discussed in CLE 036 *Engineering Change Proposals for Engineers* (DAU 2014). The steps include two meetings of the Decentralized Configuration Control Board (DCCB) and the preparation steps for each meeting. The first convening of the DCCB results in a Decision Memorandum (DM) to authorize beginning the NRE for the ECP. The second convening of the DCCB meeting reviews the test reports and other results of the ECP to approve delivery of the modified system to the fleet (NAVAIR [PMA-251] 2013).

Section 1.7 of the CMP requires an addendum to be written for new systems that do not follow the CMP.

An addendum to this CMP will be developed for each new system that does not follow the policies and procedures outlined herein. This addendum will explain the specific policies and procedures to be followed to accomplish configuration management of the item and must be approved by PMA 251. Configuration Items that follow the policies and procedures outlined herein do not require the addition of a separate addendum to this CMP. (NAVAIR [PMA-251] 2013).

IMS tasking to prepare for the two DCCB events can be determined from this resource. The DM and the DCCB approval events are key milestones and

touch other processes. Some high-level schedule logic can be determined as well:

The DM is a predecessor to local engineering processes required to change the system

Local engineering processes are predecessors to the DCCB approval

DCCB approval is a predecessor to the CVN modification

To frame the context in which these key events touch other processes, the DM allows the start of SETR entry criteria (NAVAIR 2008) preparation within the local engineering processes (OSD 2011a). Test reports from the local engineering processes are sent for DCCB approval. Finally, the CVN modification process inherits constraints from the CNAF operational availability requirements (Yardley et al. 2008).

Not all of these process steps and interactions were incorporated into the tool created in this thesis. Test events and some admin tasking remain in future work.

F. OBSOLESCENCE-DRIVEN DESIGN REFRESH PLANNING FOR SUSTAINMENT-DOMINATED SYSTEMS

Singh and Sanborn discuss a methodology for proactively and economically planning periodic refreshment of systems with shorter lifespans than the host platforms on which they are embedded. Aircraft and naval vessels are presented as examples. They discuss how, over time, the residual technological value of embedded systems decreases while the cost to re-engineer increases. Singh and Sanborn develop the “Mitigation of Obsolescence Cost Analysis” (MOCA) methodology to calculate the economically optimal refresh period (Singh and Sanborn 2006).

While the CVN availabilities are established to balance operational availability with sustainment activities (Yardley et al. 2008), Singh and Sanborn provide a method to forecast obsolescence work packages in conjunction with predicted changes to support future CSG deployment architectures. This resource spoke to the need to plan refreshes with adequate time not only to meet

the delivery date, but also to make economically based, cost-effective decisions. Employing this type of forecasting strategy on components of existing CVN IT architectures could be the impetus to begin planning obsolescence ECPs in a proactive rather than reactive manner. Using an economic forecast, rather than reacting to supportability problems, could also help a program obtain enough schedule margin to allow for an event-driven development plan called out in NAVAIRINST 4355.19D (NAVAIR 2008)

LIST OF REFERENCES

- Alberts, David S. 2011. *The Agility Advantage: A Survival Guide for Complex Enterprises and Endeavors*. Washington, DC: CCRP.
- CNAF. n.d. "Commander Naval Air Force Homepage." Accessed January 31, 2015. <http://www.public.navy.mil/AIRFOR/Pages/homepage.aspx>.
- DAU. 2014. "Continuous Learning Module CLE 036 Engineering Change Proposals for Engineers." Accessed February 5
http://icatalog.dau.mil/onlinecatalog/courses.aspx?crs_id=470.
- . 2015. "Acquipediea: Engineering Cost Estimation Method" retrieved on May 12.
<https://dap.dau.mil/acquipediea/Pages/ArticleDetails.aspx?aid=46357aef-5d79-4dd8-91b0-bb94350b8315>.
- Department of Defense. 2008. *Continuous Process Improvement/Lean Six Sigma Guidebook Revision 1*. Washington, DC: Department of Defense.
http://www.au.af.mil/au/awc/awcgate/dod/cpi_leansixsigma_hdbk2008.pdf.
- . 2011a. *Department of Defense Standard Practice Work Breakdown Structures for Defense Material Items*. MIL-STD-881C. Washington, DC: Department of Defense.
<http://www.navair.navy.mil/nawctsd/Resources/Library/Acqguide/MIL-STD%20881C%203%20Oct%2011.pdf>.
- . 2011b. "The DODAF Architecture Framework Version 2.02." Accessed November 23, 2014.
http://dodcio.defense.gov/Portals/0/Documents/DODAF/DODAF_v2-02_web.pdf.
- . 2013. "Form DD1692 (series)." Accessed January 31, 2015.
<http://www.dtic.mil/whs/directives/forms/index.htm>.
- LaGrone, Sam. 2014. USNI News. October 6.
<http://news.usni.org/2014/10/06/fleet-forces-swap-truman-eisenhower-carrier-deployments>.
- OSD. 2011a. "Systems Engineering Plan (SEP) Outline."
http://www.acq.osd.mil/se/docs/PDUSD-Approved.SEP_Outline-04-20-2011.pdf.
- . 2011b. "Systems Engineering Plan Prep Guide."
<http://www.acq.osd.mil/se/docs/SEP-Prep-Guide.pdf>.

- NASA, 2010. NASA/SP-2010-3404 Work Breakdown Structure (WBS) Handbook. Washington, D.C: National Aeronautics and Space Administration Headquarters
- Naval Air Systems Command. 2008. *Systems Engineering Technical Review Process* (NAVAIRINST 4355.19D). Patuxent River, MD: Department of the Navy (NAVAIR).
- . 2012. *Technical Directives System* (NAVAIR 00–25-300). Patuxent River, MD: Department of the Navy.
<http://navybmrc.com/study%20material/NAVAIR%2000-25-300.pdf>.
- , [4.2.3] 2010. *Integrated Government Schedule Development and Maintenance Toolkit*. Patuxent River, MD: Department of the Navy (NAVAIR [4.2.3]).
- , [PMA-251]. 2013. *PMA-251 Configuration Management Plan for Launch and Recovery Equipment* Patuxent River, MD: Department of the Navy (NAVAIR [PMA-251]).
- , [4.0]. 2015. *SWP for SWPs* (SWP4C00-001). Patuxent River, MD: Department of the Navy (NAVAIR [4.0]).
- Naval Sea Systems Command. 2004. *Configuration Management Policy and Guidance* (NAVSEAINST 4130.12B). Washington, D.C.: Department of the Navy (NAVSEA).
- NDIA. 2012. *Planning & Scheduling Excellence Guide (PASEG)*. Arlington, VA: National Defense Industrial Association.
- Singh, Pameet and Peter Sandborn. 2006. “Obsolescence Driven Design Refresh Planning for Sustainment-Dominated Systems.” *The Engineering Economist* 51, no. 2: 115–139.
- SPEC. 2013. *Using Innoslate for Program Management*. Systems and Proposal Engineering Company. Last modified on August 15, 2013.
- Tan, Hui Fang Evelyn. 2012. “Application of an Entropic Approach to Assessing Systems Integration.” Master’s thesis, Naval Postgraduate School.
- VitechCorp. 2012. “Document the Model, Don’t Model the Document.” Accessed November 10, 2014. <https://www.youtube.com/watch?v=VjGmNjg5cro>.
- . 2013a. “CORE9 Architecture Definition Guide.” Blacksburg, VA. Vitech Corporation

- . 2013b. "CORE9 System Definition Guide." Blacksburg, VA. Vitech Corporation
- . 2013c. "CORE9 Systems Engineering Guided Tour." Blacksburg, VA. Vitech Corporation
- Yali, Angela Lua, Xiao Yu Guang, Zee Sow Wai, and Loo Jang Wei. 2012. "Comprehensive Life Cycle Approach to Obsolescence Management." *DSTA Horizons*. Singapore: Defence Science and Technology Agency.
- Yardley, R.J.; James G. Kallimani, John F. Schank, and Clifford A. Grammich. 2008. *Increasing Aircraft Carrier Forward Presence: Changing the Length of the Maintenance Cycle*. Santa Monica, CA: RAND.

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