The Johns Hopkins University
APPLIED PHYSICS LABORATORY

Enclosure to: NSAD-L-2010-129
JNC04

NSAD-R-2010-037

BEST PRACTICES FOR THE DEVELOPMENT OF MODELS AND SIMULATIONS

Final Report

June 2010

The Johns Hopkins University
APPLIED PHYSICS LABORATORY

NSAD-R-2010-037

BEST PRACTICES FOR THE DEVELOPMENT OF MODELS AND SIMULATIONS

Final Report

June 2010

FOR:

Office of the Director, Defense Research and Engineering
Systems Engineering / Systems Analysis
1901 N. Beauregard St., Suite 500
Alexandria, VA 22311-1705

BY:

Johns Hopkins University - Applied Physics Laboratory
11100 Johns Hopkins Road
Laurel, MD 20723

This page intentionally left blank.

JHU/APL Team

Title	Name	Organization
Team Lead	Dr. Katherine L. Morse	JHU/APL
Program Manager	Dr. James Coolahan	JHU/APL
Team SME	Bob Lutz	JHU/APL
Team SME	Nathaniel Horner	JHU/APL
Team SME	Shon Vick	JHU/APL
Team SME	Ronda Syring	JHU/APL

This page intentionally left blank.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

EXECUTIVE SUMMARY

Although the importance and use of modeling and simulation (M&S) tools (models, simulations, and utilities) is expanding across the Department of Defense (DoD), relatively few persons have a good grasp of the process and principles that should be followed when developing such tools. In conjunction with the Institute of Electrical and Electronic Engineers (IEEE) standardization of the High Level Architecture (HLA), the Department of Defense has identified a recommended practice for federation development and execution, but no equivalent best practice exists for the development of individual modeling and simulation tools. Whether conducting such a development or overseeing a contractor's efforts to do so, DoD acquisition professionals need to understand best practices for developing modeling and simulation tools.

The Johns Hopkins University Applied Physics Laboratory (JHU/APL) was tasked to define a Systems Engineering framework focused on the development of stand-alone models and simulations, to identify practices for the efficient development and evolution of credible modeling and simulation tools, and to integrate these practices into the framework.

The study team performed a literature search to identify the major systems engineering (SE) frameworks in active use today, resulting in the following list:

1. International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) Systems engineering - System life cycle processes (ISO/IEC-15288) [Reference (a)][1].
2. IEEE Federation Development and Execution Process (IEEE 1516.3-2003)/Distributed Simulation Engineering and Execution Process (IEEE P1730) [Reference (b)].
3. American National Standards Institute (ANSI)/Electronic Industries Alliance (EIA) Processes for Engineering a System (EIA-632) [Reference (c)].
4. Institute for Electrical and Electronics Engineers Standard for Application and Management of the Systems Engineering Process (IEEE-1220) [Reference (d)].
5. Military Standard - System Engineering Management (MIL-STD-499C) [Reference (e)].
6. International Council on Systems Engineering (INCOSE) Handbook (v3.1) [Reference (f)].
7. Capability Maturity Model Integration for Development (CMMI-DEV) [Reference (g)].

Each framework was assessed to identify its applicability to the M&S domain, along with its relative strengths and weaknesses. The results of these assessments were synthesized into a new SE Framework consisting of the following phases and activities.

[1] References may be found in Appendix A.

Phase 1: *Requirements Development*

Activity 1: *Develop Stakeholder Requirements*

Activity 2: *Develop Product Requirements*

Activity 3: *Validate Requirements*

Phase 2: *Conceptual Analysis*

Activity 1: *Develop Conceptual Model*

Activity 2: *Validate Conceptual Model*

Phase 3: *Product Design*

Activity 1: *Perform Functional Analysis*

Activity 2: *Synthesize Design*

Activity 3: *Verify Design*

Phase 4: *Product Development*

Activity 1: *Establish Software Development Environment*

Activity 2: *Implement Product Design*

Phase 5: *Product Testing*

Activity 1: *Perform Product Verification*

Activity 2: *Perform Product Validation*

Project Management Practices

- *Project Planning*
- *Project Control/Resource Management*
- *Risk Management*
- *Quality Management*
- *Configuration Management*

The team performed a survey of the broadest possible audience of M&S tool developers and a literature search to identify 116 sound practices. Relying on guidance gleaned from other best practices development efforts, the team developed a list of 20 criteria for determining which sound practices qualified as best practices. After the best practices were determined according to the criteria, they were binned into the phases and activities of the SE Framework. The final result is a set of 50 best practices aligned with the SE Framework above.

The team recommends putting forward the SE Framework for standardization within the Simulation Interoperability Standards Organization to get more detailed community input into the framework and practices, and to motivate broader adoption of this work.

1 BACKGROUND

On February 1, 2005, the Department of Defense (DoD) Systems Engineering Forum, comprised of the Senior Systems Engineering Executives of the various DoD Components, chartered a subordinate body, the Acquisition Modeling and Simulation Working Group (AMSWG). The AMSWG was given four goals:

1. Recommend ways to make Modeling and Simulations (M&S) a core enabler and integral element of systems engineering (SE) processes in systems, systems of systems (SoS) and family of systems (FoS) acquisition.
2. Identify challenges to using M&S to support systems, SoS, and FoS engineering, to include test and evaluation, and make recommendations for effective, focused solutions, including revising policy.
3. Recommend ways that M&S can improve application of good SE practices.
4. Work with other organizations [such as the Defense Modeling and Simulation Office (DMSO[2])] to ensure synchronization and coordination of functional domain M&S plans.

Among the deliverables expected of the AMSWG was an Acquisition M&S Master Plan (AMSMP) [Reference (h)][3]. To meet the above goals and develop an AMSMP, the AMSWG conducted both a bottom-up review of 16 previous studies on the use of M&S in acquisition and a top-down requirements derivation from the then-current version of the Chairman of the Joint Chiefs of Staff Instruction (CJCSI) 3170.01, "Joint Capabilities Integration and Development System (JCIDS) [Reference (i)]," and DoD Directive 5000.1 "The Defense Acquisition System" [Reference (j)]. The AMSWG also benefited from several visits to field activities and briefings by government and industry SoS project managers. The briefings received from these efforts revalidated gaps identified in the top-down and bottom-up approach.

Among the gaps in M&S capabilities identified by this analysis were the following:

- Many M&S tool gaps and deficiencies exist concerning:
  - What's modeled (e.g., unconventional warfare, communication networks, threats, logistics),
  - Fidelity, granularity, interoperability, and
  - Only limited consensus on common models to be used across a domain.
- M&S developers, not M&S users, tend to drive M&S development.
- Body of knowledge for M&S support to acquisition is deficient and not managed.

[2] Now the Modeling and Simulation Coordination Office (M&S CO)
[3] References may be found in Appendix A.

- Acquisition community managers and staff are, for the most part, uninformed about M&S capabilities and limitations.
  - Weak acquisition personnel understanding of commercial M&S activities.
  - Not enough M&S specialists [no career path (except Army), no formal education or training].
- M&S developers lack understanding of modeling best practices, abstraction techniques, context dependencies, etc.

After further consideration of these gaps and potential corrective actions, the AMSWG chose to include the following action in the AMSMP published in April 2006.

ACTION 3-2. Define and foster sound practices for efficient development and evolution of credible M&S tools, incorporating user-defined requirements in a systems engineering approach with appropriate verification and validation.

The AMSMP provides the following rationale for this action:

"Although the importance and use of M&S tools (models, simulations, and utilities) is expanding across the Department of Defense, relatively few persons have a good grasp of the process and principles that should be followed when developing such tools. In conjunction with the Institute of Electrical and Electronics Engineers (IEEE) standardization of the high-level architecture (HLA), the Department of Defense has identified a recommended practice for federation development and execution, but no equivalent best practice exists for the development of individual M&S tools. Whether conducting such a development or overseeing a contractor's efforts to do so, DoD acquisition professionals need to understand best practices for developing M&S tools."

It is this AMSMP action that motivates this study.

2 STRATEGY AND APPROACH

The overall strategy for the study was to collect inputs from a broad range of recognized sources, and synthesize the inputs into a Systems Engineering Framework with integrated Best Practices.

The SE Framework was developed based on a side-by-side analysis of the major phases of seven widely recognized systems engineering frameworks and processes. Only one of these extant frameworks, the Federation Development and Execution Process (FEDEP) [Reference (b)], focuses expressly on M&S, so the FEDEP formed the basis of M&S-specific guidance for the SE Framework. This ensures that the SE Framework considers M&S particular considerations such as conceptual modeling and time management; obtaining and transforming authoritative data; and verification, validation and accreditation (VV&A).

Individual practices were collected from two types of sources:

- A literature search of relevant books, journals, and conferences, and
- A survey targeted at the broadest possible audience of M&S tool developers in industry, government, and academia.

The team developed a list of criteria for assessing whether the identified practices qualified as "best practices." Practices that passed this filter were assigned by consensus of the team into the phases of the SE Framework. Along the way, progress on the study was reported at several conferences and meetings.

Finally, the draft SE Framework and Best Practices were reviewed by organizations and individuals that provided inputs to ensure the correctness and appropriateness of the Best Practices.

2.1 MAJOR SYSTEMS ENGINEERING FRAMEWORKS AND PROCESSES

The team began by reviewing the major SE frameworks in active use today:

1. International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) Systems engineering - System life cycle processes (ISO/IEC-15288) [Reference (a)].
2. IEEE Federation Development and Execution Process (IEEE 1516.3-2003)/Distributed Simulation Engineering and Execution Process (IEEE P1730) [Reference (b)].
3. American National Standards Institute (ANSI)/Electronic Industries Alliance (EIA) Processes for Engineering a System (EIA-632) [Reference (c)].
4. Institute for Electrical and Electronics Engineers Standard for Application and Management of the Systems Engineering Process (IEEE-1220) [Reference (d].

5. Military Standard - System Engineering Management (MIL-STD-499C) [Reference (e)].
6. International Council on Systems Engineering (INCOSE) Handbook (v3.1) [Reference (f)].
7. Capability Maturity Model Integration for Development (CMMI-DEV) [Reference (g)].

For each such review, an assessment was made to identify the applicability of each framework to the M&S domain, along with its relative strengths and weaknesses. These assessments include substantial quotes and paraphrases from the frameworks themselves. Then, starting in Section 2.2, the strengths and weaknesses of each framework are compared across the range of activities needed to build stand-alone M&S applications, and selections made (along with supporting rationale) for the content and organization of the Models and Simulations Development Best Practices (MSDBP) SE Framework. Finally, in Section 3, these selections are summarized in a graphical form, along with textual descriptions of all major phases and associated development activities defined in the SE Framework.

2.1.1 ISO/IEC-15288 – Systems and Software Engineering - System Life Cycle Processes

2.1.1.1 Summary

The purpose of the ISO/IEC 15288 standard, as identified in the standard itself, is to "establish a common process framework for describing the life cycle of man-made systems". The purpose of the standard is further articulated as "a set of processes and associated terminology for the full life cycle, including conception, development, production, utilization, support and retirement" [Reference (a)]. The standard supports the definition, control, assessment, and improvement of these processes. Note that the IEEE Computer Society collaborated with ISO/IEC Joint Technical Committee (JTC) 1 in the development of the 15288 standard, and thus owns a joint copyright to this material.

In terms of applicability to potential user groups, the standard states "there is a wide variety of systems in terms of their purpose, domain of application, complexity, size, novelty, adaptability, quantities, locations, life spans and evolution. This international standard describes the processes that comprise the life cycle of any man-made system" [Reference (a)]. The standard goes on to say that "it therefore applies to one-of-a-kind systems, mass-produced systems and customized, adaptable systems. It also applies to a complete stand-alone system and to systems that are embedded and integrated into larger more complex and complete systems" [Reference (a)]. Since this standard defines a process framework applicable to any type of system (including M&S systems) and specifically identifies its applicability to stand-alone systems, it appears to provide a reasonable candidate for the MSDBP SE Framework.

Crucial to the 15288 standard is the concept of a "life cycle" model. The standard defines a life cycle *model* to be a "framework of processes and activities concerned with the life cycle that may be organized into stages, which also acts as a common reference for communication and understanding". The life cycle model defined in the 15288 standard is shown in Figure 2-1. In this organization, there are a total of four different process groups:

> **Agreement Processes**: These processes are concerned with ensuring the establishment of agreements among participating organizations. One organization (acting as an acquirer) can task another (acting as a supplier) for products or services using these agreements.
>
> **Organizational Project-Enabling Processes**: These processes are concerned with ensuring that the resources needed to enable the project to meet the needs and expectations of the organization's interested parties are met. The Organizational Project-Enabling Processes are typically concerned at a strategic level with the management and improvement of the organization's business or undertaking, with the provision and deployment of resources and assets, and with its management of risks in competitive or uncertain situations.
>
> **Project Processes**: These processes are concerned with managing the resources and assets allocated by organization management and with applying them to fulfill the agreements into which the organization or organizations enter. They relate to the management of projects, in particular to planning in terms of cost, timescales and achievements, to the checking of actions to ensure that they comply with plans and performance criteria and to the identification and selection of corrective actions that recover shortfalls in progress and achievement.
>
> **Technical Processes**: These processes are concerned with technical actions throughout the life cycle. They transform the needs of stakeholders first into a product and then, by applying that product, provide a sustainable service, when and where needed in order to achieve customer satisfaction. The Technical Processes are applied in order to create and use a system, whether it is in the form of a model or is a finished product, and they apply at any level in a hierarchy of system structure.

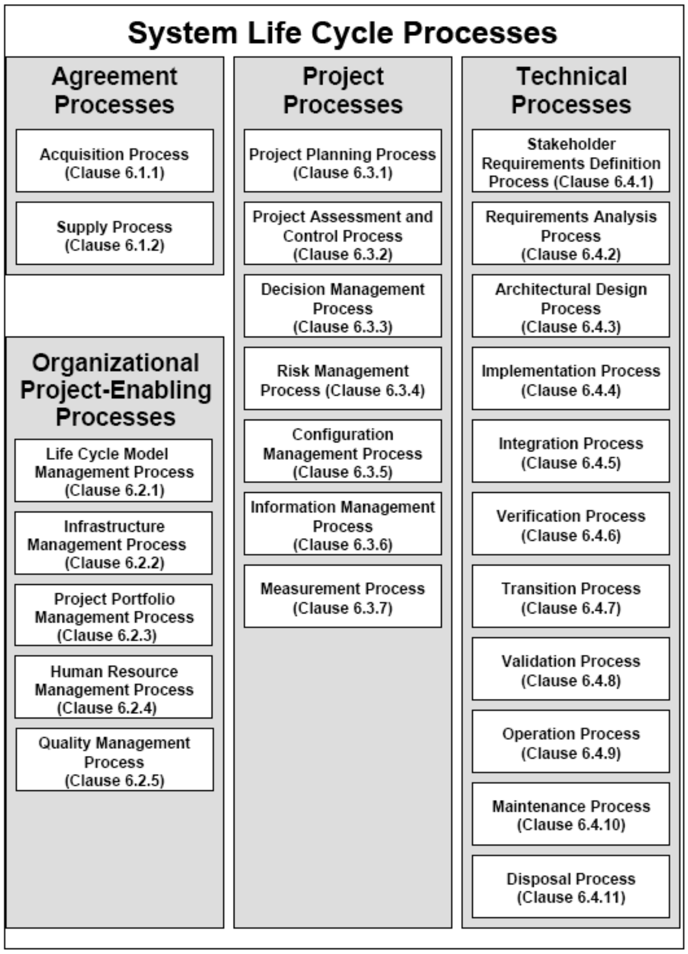

Figure 2-1: ISO/IEC 15288 SE Processes

For the MSDBP SE Framework, the main process group of interest is the Technical Processes. A short description of each process in this group is provided below:

Stakeholder Requirements Definition Process: defines the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment.

Requirements Analysis Process: transforms the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services.

Architectural Design Process: synthesizes a solution that satisfies system requirements. This process encapsulates and defines areas of solution expressed as a set of separate problems of manageable, conceptual and, ultimately, realizable proportions.

Implementation Process: realizes a specified system element. This process transforms specified behavior, interfaces and implementation constraints into fabrication actions that create a system element according to the practices of the selected implementation technology.

Integration Process: assembles a system that is consistent with the architectural design. This process combines system elements to form complete or partial system configurations in order to create a product specified in the system requirements.

Verification Process: confirms that the specified design requirements are fulfilled by the system. This process provides the information required to effect the remedial actions that correct non-conformances in the realized system or the processes that act on it.

Transition Process: establishes a capability to provide services specified by stakeholder requirements in the operational environment. This process installs a verified system, together with relevant enabling systems, e.g., operating system, support system, operator training system, user-training system, as defined in agreements.

Validation Process: provides objective evidence that the services provided by a system when in use comply with stakeholders' requirements, achieving its intended use in its intended operational environment.

Operation Process: uses the system in order to deliver its services. This process assigns personnel to operate the system, and monitors the services and operator-system performance.

Maintenance Process: sustains the capability of the system to provide a service. This process monitors the system's capability to deliver services, records problems for analysis, takes corrective, adaptive, perfective and preventive actions and confirms restored capability.

Disposal Process: ends the existence of a system entity. This process deactivates, disassembles and removes the system and any waste products, consigning them to a final condition and returning the environment to its original or an acceptable condition.

Obviously, the other process groups also reflect issues of interest to developers of stand-alone M&S tools. This is especially true in the Project Processes group, in that concerns such as project planning, risk management, and configuration management are all areas in which M&S best practices can be captured.

All of the process descriptions in ISO/IEC 15288 include a set of lower-level activities and tasks that are needed to implement the defined process successfully. ISO/IEC 15288 also includes a set of supporting annexes that provides additional guidance to the implementer of the standard. Examples of the annexes include a discussion of how to tailor the process to satisfy particular circumstances or factors, a specialized process view for specialty engineering, and relationships with other IEEE standards.

2.1.1.2 Applicability of ISO/IEC 15288 to M&S Development

Many of the processes, activities, and tasks described in the ISO/IEC 15288 standard are directly applicable to stand-alone M&S development. The organization and sequencing of the various processes (as well as the lower-level activities/tasks) in the Technical Processes group is quite similar to the IEEE 1516.3 High Level Architecture Federation Development and

Execution Process standard for M&S development. However, while the focus of the FEDEP is on distributed M&S, the ISO/IEC 15288 standard identifies some additional technical and managerial considerations applicable for either stand-alone or distributed M&S.

While the ISO/IEC 15288 standard identifies the processes, activities, and tasks for engineering a system, the issue of activity/task sequencing is not discussed in the standard. This could be considered a deficiency, in that users could assume that the overall process must be performed in the exact order defined in the process descriptions. It is assumed that this is not the intent, and in fact, there is a "tailoring" annex that shows how the process can be adapted for different uses.

Conformance is another area in which ISO/IEC 15288 is not an exact fit for the needs of the MSDBP project. This document is a formal standard, and defines many hard process requirements (i.e., the imperative "shall" is used throughout the document) for which conformance rules are defined. The SE Framework for MSDBP is envisioned as general guidance for stand-alone M&S developers (i.e., a guidance document rather than a formal standard) and, thus, precludes strict conformance rules or any uses of the word “shall.” This must be taken into account when considering the reuse potential of this process framework for MSDBP.

The scope of the SE process defined in this document is also somewhat different from what is desirable for the MSDBP project. ISO/IEC 15288 is concerned with all aspects of the system life cycle process, from initial requirements development through disposal. In addition, ISO/IEC 15288 also includes all supporting life cycle processes in such areas as acquisition/supply, project management, human resources, project control, and others. Again, although these processes are all important systems engineering considerations, the envisioned MSDBP SE Framework will have a narrower scope. Specifically, the MSDBP SE Framework will include all the technical processes needed to develop the desired product, along with supporting processes as needed to enable/control technical activities. Thus, longer-term technical life cycle processes such as maintenance and disposal are likely to be out of scope, as are certain administrative/managerial processes that do not directly concern the process of M&S development (e.g., personnel availability). The determination of the exact scope of the MSDBP SE Framework will drive exactly what ISO/IEC 15288 processes are the best candidates for SE Framework incorporation.

2.1.2 IEEE 1516.3 – High Level Architecture Federation Development and Execution Process

2.1.2.1 Summary

The purpose of the IEEE 1516.3 (FEDEP) recommended practice, as identified in the document itself, is to "define the processes and procedures that should be followed by users of the High Level Architecture to develop and execute federations". It also states that "the FEDEP is not intended to replace low-level management and systems engineering practices native to HLA user organizations, but is rather intended as a higher-level framework into which such practices can be integrated and tailored for specific uses". Since the FEDEP is a SE framework that was specifically designed for the M&S domain, it appears to provide a suitable foundation for the MSDBP SE Framework even though its focus is primarily distributed systems.

The FEDEP is organized according to a series of seven major steps. These steps are illustrated in Figure 2-2.

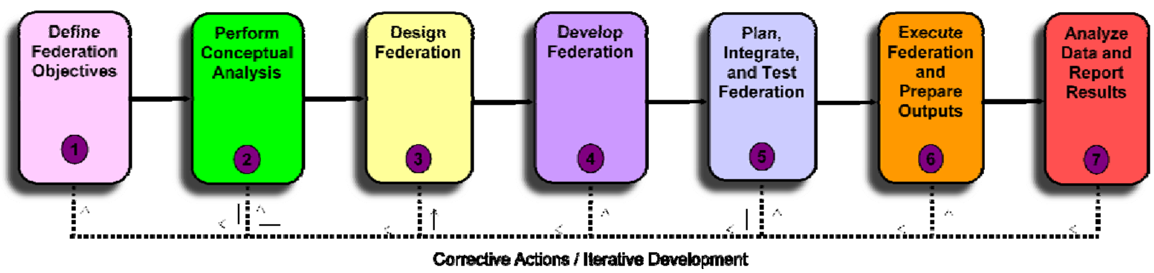

Figure 2-2: HLA FEDEP High-Level View

A short description of each major step is as follows:

Step 1: *Define federation objectives:* The federation user, the sponsor, and the federation development team define and agree on a set of objectives and document what must be accomplished to achieve those objectives.

Step 2: *Perform conceptual analysis:* Based on the characteristics of the problem space, an appropriate representation of the real world domain is developed.

Step 3: *Design federation:* Existing federates that are suitable for reuse are identified, design activities for federate modifications and/or new federates are performed, required functionalities are allocated to the federates, and a plan is developed for federation development and implementation.

Step 4: *Develop federation:* The Federation Object Model (FOM) is developed, federate agreements are established, and new federates and/or modifications to existing federates are implemented.

Step 5: *Plan, integrate, and test federation:* All necessary federation integration activities are performed, and testing is conducted to ensure that interoperability requirements are being met.

Step 6: *Execute federation and prepare outputs:* The federation is executed and the output data from the federation execution is pre-processed.

Step 7*: Analyze data and evaluate results:* The output data from the federation execution is analyzed and evaluated, and results are reported back to the user/sponsor.

In the FEDEP document, each major step is decomposed into a set of interrelated activities. Each activity is further characterized by its inputs, outcomes, and lower-level recommended tasks needed to perform the activity successfully. The FEDEP also describes the flow of information from activity to activity both within and across major steps, and describes the interim products that are produced as a result of each activity and task.

In terms of activity sequencing, the FEDEP specifically states that many activities are performed cyclically and/or concurrently, and that users should not feel restricted to implementing the various activities in the order presented in the document. Instead, users should tailor the development process to fit the needs of their specific programs. This use of general user guidance rather than hard user requirements is indicative of most recommended practice documents.

2.1.2.2 Applicability to M&S Development

The FEDEP has much to offer as an SE framework for MSDBP. First, it provides an SE framework specifically tailored to the M&S domain, which is unique among the SE process references considered for this effort. This framework covers the full range of activities of interest, from initial requirements development to post-execution analysis. It is also a recommended practice rather than a standard, implying that it is written in the same style in which an SE framework for MSDBP would need to be provided. Further, since the FEDEP is established and widely recognized within the M&S community, there may be benefits in using the same (or similar) framework for stand-alone M&S development as is used for distributed M&S development.

Despite its benefits, there are clearly some aspects of the FEDEP that could not be reused for MSDBP. The obvious difference is its targeted user community. The FEDEP is written specifically for distributed simulation, and thus includes a variety of activities that are not relevant to developers of stand-alone M&S tools. For instance, activities like "Select Federates," "Develop FOM," and "Integrate Federation" are generally not relevant to stand-alone M&S tool developers, while certain activities that are of interest to stand-alone M&S tool developers (such as event queue management and user interface design) are absent in the FEDEP. In many ways, the existence of a development process for individual M&S tools is simply assumed in the FEDEP, with the FEDEP focusing instead on how to integrate these individual M&S tools into larger distributed M&S environments.

Although the seven higher-order steps are generally applicable to either stand-alone or distributed M&S systems, some of these steps may not be relevant to the MSDBP SE Framework. The issue is one of scope. The FEDEP covers the process of development, execution, and post-execution analysis, while the MSDBP Framework is largely focused on development only. Thus, some of the later stages of the FEDEP may be considered out of scope. The analysis of the potential contributions of all of the reference SE processes will consider the boundaries where best practices are to be defined, and adjust the scope of the SE framework accordingly.

2.1.3 ANSI/EIA-632 – Processes for Engineering a System

2.1.3.1 Summary

EIA-632 was developed as a joint project of the EIA and the International Council on Systems Engineering. The purpose of the EIA-632 standard, as identified in the standard itself, is to "provide an integrated set of fundamental processes to aid a developer in the engineering or reengineering of a system." The standard also indicates that "use of this standard is intended to help developers:

- Establish and evolve a complete and consistent set of requirements that will enable delivery of feasible and cost-effective system solutions.
- Satisfy requirements within cost, schedule, and risk constraints.
- Provide a system, or any portion of a system, that satisfies stakeholders over the life of the products that make up the system.
- Provide for the safe and/or cost-effective disposal or retirement of a system."

Since one of the intended uses of the standard is to “develop lower-tier industry- or domain-specific process standards,” it could potentially provide a viable SE Framework for this project.

EIA-632 identifies several processes for engineering a system, which are organized into five groups as shown in Figure 2-3. A short description of each group is provided below.

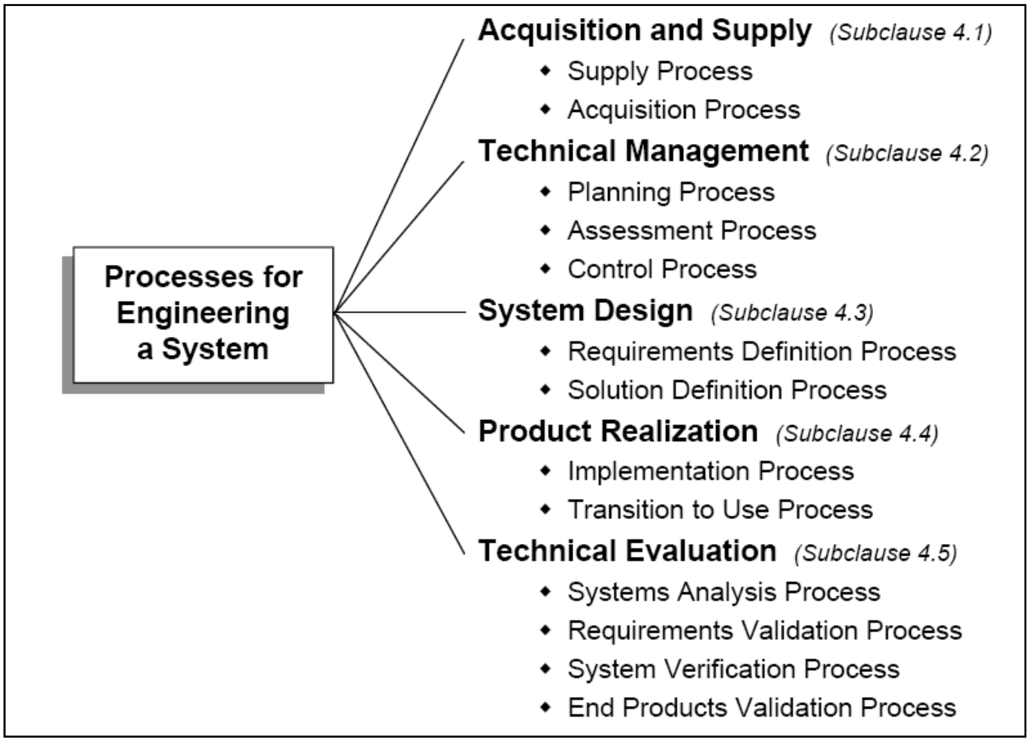

Figure 2-3: EIA-632 SE Processes

Acquisition and Supply - The Acquisition and Supply processes are used by a developer to arrive at an agreement with another party to accomplish specific work and to deliver required products, or with another party or parties to have work done and to obtain desired products. A short description of each process follows:

Supply: The process used by the developer (when acting as a supplier) to establish and satisfy an agreement with the acquirer.

Acquisition: The process used by the developer (when acting as an acquirer) to establish an agreement with a supplier and to manage supplier performance.

Technical Management - The Technical Management processes are to be used to plan, assess, and control the technical work efforts required to satisfy the established agreement. This group consists of the following three processes:

Planning: This process is used to support enterprise and project decision-making and to prepare necessary technical plans that support and complement project plans.

Assessment: This process is used to (1) determine progress of the technical effort against both plans and requirements; (2) review progress during technical reviews; and (3) support control of the engineering of a system.

Control: This process is used to (1) manage the conduct and outcomes of the Acquisition and Supply Processes, System Design Processes, Planning and Assessment

Processes, Product Realization Processes, and Technical Evaluation Processes; (2) monitor variations from the plan and anomalies relative to requirements; (3) distribute required and requested information; and (4) ensure necessary communications.

System Design - The System Design processes are used to convert agreed-upon requirements of the acquirer into a set of realizable products that satisfy acquirer and other stakeholder requirements. This group consists of the following two processes:

Requirements Definition - This process is used to: (1) identify, collect, and define acquirer and other stakeholder requirements and (2) transform these requirements into a set of validated system technical requirements.

Solution Definition - This process is used to transform validated system technical requirements into an acceptable design solution.

Product Realization - The Product Realization Processes are used to: (1) convert the specified requirements and other design solution characterizations into either a verified end product or a set of end products in accordance with the agreement and other stakeholder requirements; (2) deliver these to designated operating, customer, or storage sites; (3) install these at designated operating sites or into designated platforms; and (4) provide in-service support, as called for in an agreement. This group consists of the following two processes:

Implementation: This process is used to transform the characterized design solution into an integrated end product that conforms to its specified requirements.

Transition to Use: This process results in products delivered to the appropriate destinations, in the required condition for use by the acquirer, and for the appropriate training of installers, operators, or maintainers of the products.

Technical Evaluation: The Technical Evaluation Processes are intended to be invoked by one of the other processes for engineering a system. This group consists of the following four processes:

Systems Analysis: This process is used to (1) provide a rigorous basis for technical decision making, resolution of requirement conflicts, and assessment of alternative physical solutions; (2) determine progress in satisfying system technical and derived technical requirements; (3) support risk management; and (4) ensure that decisions are made only after evaluating the cost, schedule, performance, and risk effects on the engineering or reengineering of the system.

Requirements Validation: This process is used to ensure that the requirements are necessary and sufficient for creating design solutions appropriate to meeting the exit criteria of the applicable engineering life cycle phase and of the enterprise-based life cycle phase in which the engineering or reengineering efforts occur.

System Verification: This process is used to ascertain that (1) the system design solution is consistent with its source requirements, (2) end products at each level of the system structure implementation meet their specified requirements, (3) enabling product development or procurement for each associated process is properly progressing, and (4) required enabling products will be ready and available when needed to perform.

End Products Validation: This process is used to demonstrate that the products to be delivered, or that have been delivered, satisfy the validated acquirer requirements (for example, customer, user, or operator requirements, or assigned requirements) that were input to the system design processes and that are applicable to the resulting end products.

All of the process descriptions in EIA-632 are supplemented with a set of associated requirements and lower-level tasks, which provides considerable insight into the conduct of each process. EIA-632 also includes a set of supporting annexes that provides additional guidance to the implementer of the standard. Examples of the annexes include a listing of typical planning documents, a set of process task outcomes, and a glossary of terms.

2.1.3.2 Applicability to M&S Development

Since M&S tools are a specific kind of system, and EIA-632 is a general standard, much of the standard is relevant to use in the SE Framework. Certainly, from a purely technical perspective, both of the core processes in the System Design group are directly applicable to M&S, as is the "Implementation" process within the Product Realization group. Also, the various verification and validation processes in the Technical Evaluation group overlay many development activities inherent to M&S tool development. The “Systems Analysis” process in this same group is also largely M&S-relevant, in that many of the lower-level tasks in this process reflect the fundamental design trade-off and risk analysis/mitigation activities needed in any M&S development.

Another process that is very relevant to M&S development is the "Transition to Use" process within the Product Realization group. The types of tasks described in this process are extremely important for ensuring the usability of the M&S product, but are not properly emphasized in many other process descriptions. Examples include user training, installation at customer sites, and product maintenance.

There are also some aspects of this standard that will not be applicable to the SE Framework for M&S development. For instance, most of the processes defined in the Acquisition and Supply and the Technical Management groups are out of scope for the M&S Best Practices effort, since they deal with practical management and control issues that transcend the interests of the intended user of this document (i.e., M&S developers). This is not to say that these issues are of little importance, but rather to say that such issues are best addressed in other views of the M&S development process (e.g., project manager's view, contract administrator's view), since attempting to include all possible user perspectives in a single document will result in a heavyweight process description that will be overly complex and unnecessarily wordy for any individual user.

2.1.4 ISO/IEC 26702 – Systems Engineering – Application and Management of the Systems Engineering Process (IEEE 1220)

2.1.4.1 Summary

The ISO/IEC 26702 standard defines "the interdisciplinary tasks that are required throughout a system's life cycle to transform stakeholder needs, requirements, and constraints into a system solution." A more complete articulation of the purpose is provided in the document's Introduction section:

"This standard defines the requirements for an enterprise's total technical effort related to development of products (including computers and software) and processes that will provide life cycle support (sustain and evolve) for the products. It prescribes an integrated technical approach to engineering a system and requires the application and management of the systems engineering process throughout a product life cycle. The systems engineering process is applied recursively to the development or incremental improvement of a product to satisfy market requirements and to simultaneously provide related life cycle processes for product development, manufacturing, test, distribution, operation, support, training, and disposal" [Reference (d)].

This standard is also supported by the IEEE as *IEEE Standard 1220*. ISO/IEC 26702 was prepared and adopted under a special "fast-track procedure" by ISO/IEC Joint Technical Committee (JTC) 1. The IEEE Computer Society cooperates in the maintenance of this standard as a Category A[4] liaison to Subcommittee (SC) 7.

The core content of the ISO/IEC 26702 standard is provided in three main sections:

- Clause 4 establishes requirements for planning and implementing an effective systems engineering capability within an enterprise.
- Clause 5 provides a description of the application of the Systems Engineering Process (SEP) through system definition, subsystem definition, production, and support.
- Clause 6 provides the detailed tasks of the SEP to be tailored and performed to develop product solutions and their supporting life cycle processes.

There is a close relationship among these three sections. Clause 4 lists the general requirements that a project or enterprise must accomplish to produce a total system solution. In

[4] Organizations that make an effective contribution to the work of the technical committee or subcommittee for questions dealt with by this technical committee or subcommittee. Such organizations are sent copies of all relevant documentation and are invited to meetings. They may nominate experts to participate in a Working Group/Project Team. <http://standards.ieee.org/intl/cp_establish.html>

Clause 5, the project/enterprise implements requirements of Clause 4 by applying the SEP tasks of Clause 6, as appropriate, during each life cycle stage, to evolve system details and resolve reported problems. Clause 5 is where system life cycle stages are defined, which is of most interest for defining the MSDBP SE Framework.

Clause 5 defines two main stages of development, that of *System Definition* and *Subsystem Definition*. The Subsystem Definition stage is further decomposed into three steps of development, that of *Preliminary Design, Detailed Design, and Fabrication, Assembly, Integration, and Test* (FAIT). The SEP, as described in Clause 6, is applied recursively during each stage of system development (system, subsystem, and component) to evolve and mature the product. Clause 5 also defines two main stages of operations, that of *Production* and *Support*. The SEP is applied during these stages (as well as during FAIT) to resolve reported problems and to evolve products to improve performance or extend service life. The major system life cycle stages are summarized in Figure 2-4. Short descriptions of each major stage/step are provided below.

<table>
<tr><td rowspan="2">System definition</td><td colspan="3">Subsystem definition</td><td rowspan="2">Production
Support</td></tr>
<tr><td>Preliminary design</td><td>Detailed design</td><td>Fabrication assembly, integration, and test (FAIT)</td></tr>
</table>

Figure 2-4: ISO/IEC 26702 System Life Cycle Stages

System Definition: Establishes the definition of the system with a focus on system products required to satisfy operational requirements. The major events of this stage should include completion of system, product, and subsystem interface specifications, system and product specifications, and preliminary subsystem specifications; establishment of a system baseline; and completion of technical reviews appropriate to the system definition stage. The technical reviews should evaluate the maturity of the system development and the readiness to progress to subsystem definition.

Preliminary Design: Initiates subsystem design and creates subsystem-level specifications and design-to baselines to guide component development. The project applies the SEP for the purpose of decomposing identified subsystem functions into lower-level functions and allocating functional and performance requirements to component-level functional and physical architectures.

Detailed Design: Completes subsystem design down to the lowest component level and creates a component specification and build-to component baseline for each component. The outputs of this stage are used to guide fabrication of preproduction prototypes for development test. The SEP is applied as many times as needed to decompose identified component functions into lower-level functions, and allocate functional and performance requirements throughout the resulting lower-level functional and design architectures.

Fabrication, Assembly, Integration, and Test: Resolves product deficiencies when specifications for the system, product, subsystem, assembly, or component are not met, as determined by inspection, analysis, demonstration, or test. The purpose of the FAIT stage of subsystem definition is to verify that the products designed satisfy specifications.

Production and Support: Corrects deficiencies discovered during production, assembly, integration, and acceptance testing of products and/or life cycle process products. The SEP is applied during support to evolve the product to implement an incremental change, resolve product or service deficiencies, or to implement planned evolutionary growth.

The SEP itself is summarized in Figure 2-5. A short summary of each major phase of this process is provided below.

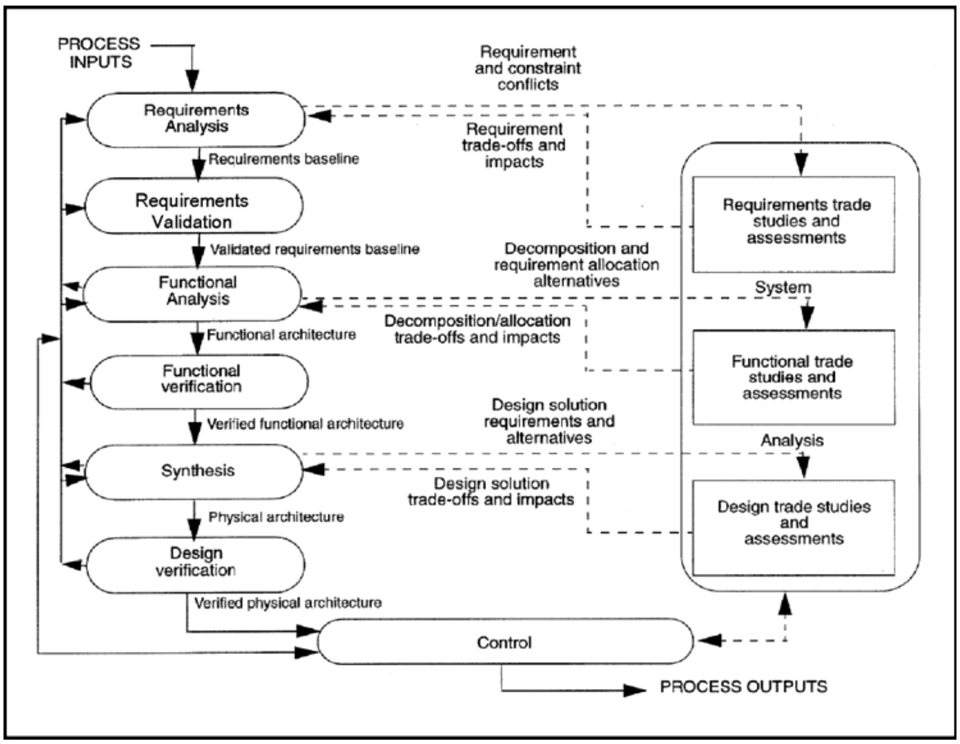

Figure 2-5: ISO/IEC 26702 Systems Engineering Process

Requirements Analysis: Establishes what the system will be capable of accomplishing; how well system products are to perform in quantitative, measurable terms; the environments in which system products operate; the requirements of the human/system interfaces; the physical/aesthetic characteristics; and constraints that affect design solutions.

Requirements Validation: Evaluates the requirements baseline that was established during requirements analysis to ensure that it represents identified stakeholder expectations and project, enterprise, and external constraints. In addition, the requirements baseline is assessed to determine whether the full spectrum of possible system operations and system life cycle support concepts has been adequately addressed.

Functional Analysis: Describes the problem defined by requirements analysis in clearer detail, and decomposes the system functions to lower-level functions that should be satisfied by elements of the system design (e.g., subsystems, components, or parts)

Functional Verification: Assesses the completeness of the functional architecture in satisfying the validated requirements baseline and to produce a verified functional architecture for input to synthesis.

Synthesis: Defines design solutions and identifies subsystems to satisfy the requirements of the verified functional architecture. Synthesis translates the functional architecture into a design architecture that provides an arrangement of system elements, their decomposition, interfaces (internal and external), and design constraints.

Design Verification: Assures that the requirements of the lowest level of the design architecture, including derived requirements, are traceable to the verified functional architecture. Also ensures that the design architecture satisfies the validated requirements baseline.

Systems Analysis: Resolves conflicts identified during requirements analysis, decomposing functional requirements and allocating performance requirements during functional analysis, evaluating the effectiveness of alternative design solutions and selecting the best design solution during synthesis, assessing system effectiveness, and managing risk factors throughout the systems engineering effort.

Control: Provides technical management and documents the activities of the SEP. This includes control of data and configuration of the design solutions, interfaces, risks, and technical progress.

ISO/IEC 26702 also provides several annexes. Examples include a description of the role of systems engineering within an enterprise, a template for the systems engineering management plan, and a description of how to use the ISO/IEC 26702 standard in conjunction with the ISO/IEC standard 15288.

2.1.4.2 Applicability to M&S Development

Like most standard systems engineering process descriptions, much of what is described in this document is applicable to M&S systems. That is, successful M&S tool development efforts do need to follow a process much like other software (and even hardware) systems, where a set of requirements drives the development of a system concept, which then progresses through several design and development iterations until the tool fully meets the stated requirements. In fact, the strong emphasis in this document on multiple spirals of a core systems engineering process as the product moves through its various life cycle stages is generally the way M&S tool development occurs in practice. Thus, there is much in this document that could be leveraged in creating an SE Framework for MSDBP.

There are also a number of characteristics of this process that should not be carried forward into the MSDBP SE Framework. One example is that, like ISO/IEC 15288, there are many lower-level life cycle processes that are out of scope for the goals of the MSDBP SE Framework. That is, while issues such as human factors, supportability, and disposal are clearly important, such issues are clearly not the focus of an SE Framework specifically tailored to M&S tool development. ISO/IEC 26702 is also a rather heavyweight process, explicitly identifying many activities and interim products that are of little concern to M&S developers (especially in small, nimble companies). In fact, this document seems to be mainly focused on the engineering of hardware systems, where manufacturing and fabrication issues such as parts inventory, supplier-vendor relationships, and safety procedures get considerable emphasis.

The organization of the ISO/IEC 26702 document was also somewhat confusing to the study team. Although the document did include a short discussion of the relationship between the three major clauses, it wasn't clear exactly how the different sections were to be used in tandem. For instance, although it was stated that the SEP described in Clause 6 was to be used in each of the Clause 5 life cycle stages, it was less clear how the implementation of the SEP changes as the product progresses from stage to stage. Also, while M&S is identified as a means of performing analysis and supporting decision-making in Clause 4, it was not clear exactly how M&S fits into the various SEP iterations. In general, the document could have done a better job integrating the different clauses into a single logical systems engineering flow.

There is a considerable amount of overlap between ISO/IEC 26702 and ISO/IEC 15288. The annex (Annex C) that is provided in the ISO/IEC standard (which compares these two documents) was quite useful in showing this relationship. Since the activities and processes described in these documents are clearly consistent, and because (in the opinion of the study team) the ISO/IEC 15288 document has superior organization and clarity of description, it is questionable just what the ISO/IEC 26702 standard could contribute to the MSDBP SE Framework beyond what ISO/IEC 15288 already provides. As stated earlier, this notion of an underlying SEP that is applied iteratively across multiple life cycle stages seems to be the main contribution of this particular standard.

2.1.5 MIL-STD-499C – Systems Engineering

2.1.5.1 Summary

The purpose of the MIL-STD-499C standard is to "describe and require a disciplined systems engineering approach in system acquisitions" [Reference (e)]. MIL-STD-499C is a compliance document, specifically intended to define systems engineering requirements for DoD contractors. The objectives of the standard are to define the minimum essential work products, produced in the systems engineering process, needed to:

- Adequately define a system over its life cycle such that the integrated system, when deployed, provides at least the threshold or minimum needed capabilities and is affordable, but otherwise balances capability, cost, schedule, risk, and the potential for evolutionary growth.
- Define clear-cut intermediate development stages to be used by the tasking and performing activities to plan, monitor, and control the progress over each phase and contract of the system acquisition program such that the first objective is achieved effectively and efficiently.

The organization of the document centers on a core Systems Engineering Process that is performed in an iterative fashion throughout the system life cycle. Figure 2-6 provides a high-level view of this process:

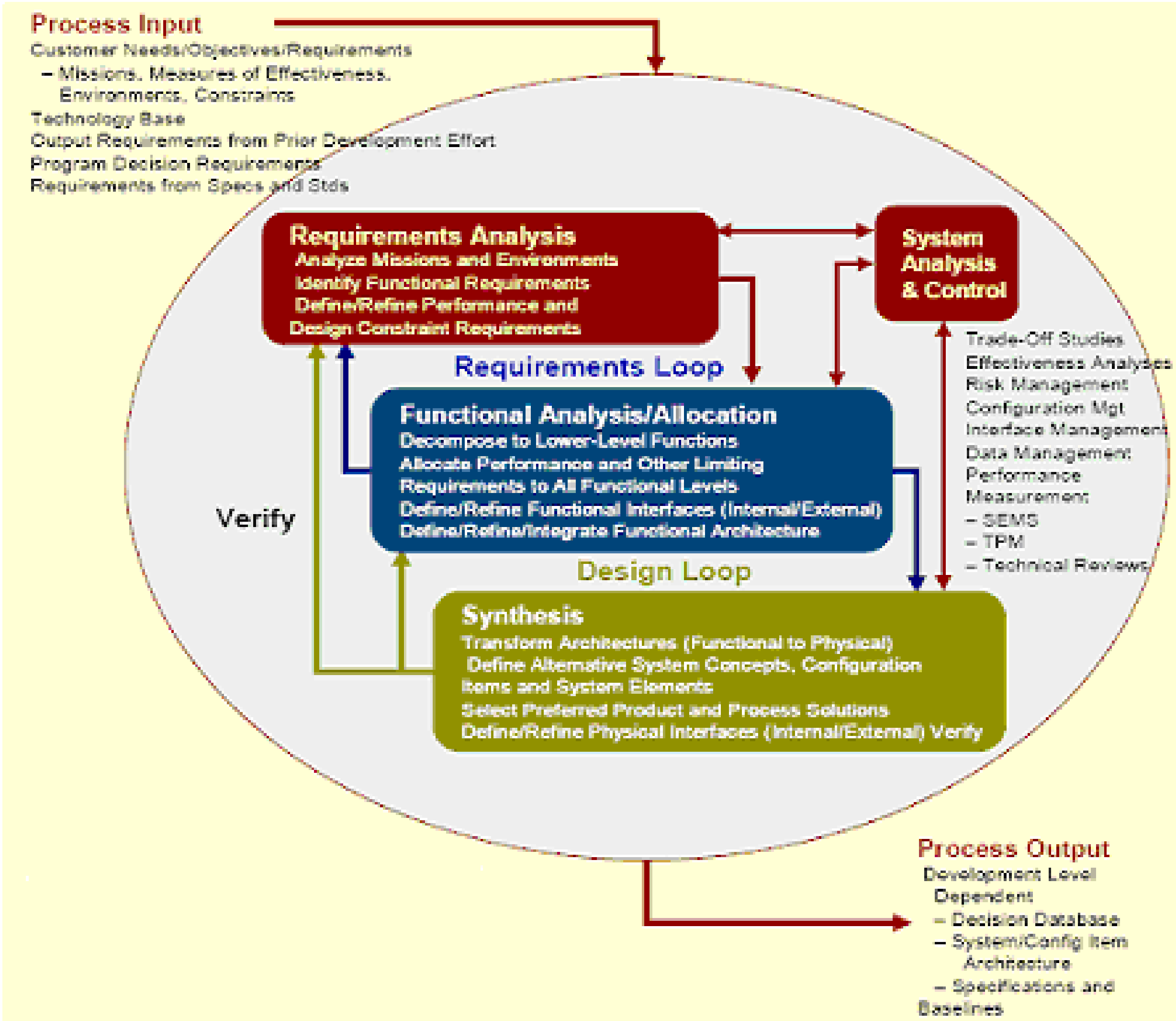

Figure 2-6: MIL-STD-499C Systems Engineering Process

Figure 2-7 relates the activities of Figure 2-6 to the evolving requirements, allocated, design release, and product configuration baselines:

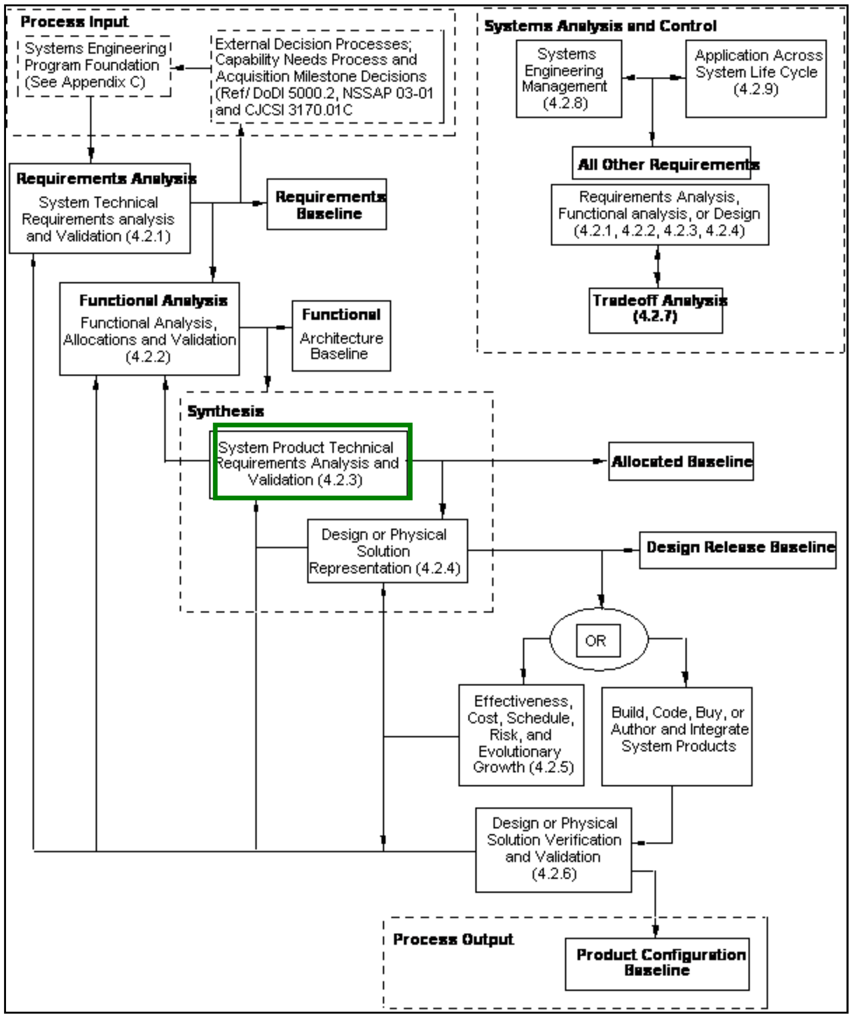

Figure 2-7: Relationship between SEP and Different Baselines

In the MIL-STD-499C document, a short description of each SE activity is provided. In addition, the products produced by each SE activity are identified along with explicit characteristics of each product. For instance, the Allocated Baseline shown as the product of Activity 4.2.3 (System Product Technical Requirements Analysis and Validation outlined in green in Figure 2-7) is characterized as follows:

- Include the physical hierarchy that identifies all system products, and shall establish the interactions of the system.
- Include the design-to technical functional and performance requirements and design constraints for each product in the physical hierarchy allocated such that requirements baselines will be fully satisfied over the system life cycle.

- Include all derived design-to requirements and design constraints for each product in the physical hierarchy.
- Include all interfaces that shall be defined at the earliest possible time and to as great a detail as is possible. In addition, in defining interfaces, the Contractor shall address how the interface will be physically implemented, as well as the logical issues such as data formats, data semantics, etc.
- Include a verification method of analysis, inspection, demonstration, or test selected for each requirement and constraint.

It is emphasized in this document that the SEP is to be applied iteratively across all life cycle stages. The evolution of the baselines as the SEP is applied during each stage is illustrated in Table 2-1.

Table 2-1: MIL-STD-499C Life Cycle Phases

DoDI 5000.2 Phase	Requirements Baseline	Allocated Baseline	Design Release Baseline
Concept Refinement	Preliminary, focus on support to JCIDS	Preliminary, focus on physical elements which drive cost, risk, and other considerations	Preliminary – basis for support to capability needs process and for concept refinement
Technology Development	Draft which balances system effectiveness, cost, schedule, risk, and growth potential	Preliminary, focus on physical ele-ments which drive risk or other con-siderations	Preliminary – reflects concept refinement and basis for technology maturation and other risk reduction
	Approved	Draft which bal-ances system effectiveness, cost, schedule, risk and growth potential	Preliminary – basis for technology selection and for the assessment to support requirements baseline validation
System Integration	Maintained	Approved	Draft – basis for assessment to support allocated baseline validation
	Maintained	Maintained	Approved - build, buy, code, author, and integrate developmental system products for qualification
System Demo	Maintained	Maintained	Maintained
	Maintained	Maintained	Maintained
Production and Deployment	Maintained	Maintained	–

Clause 5 of the document adds additional detail to contractor requirements across several specialty areas, such as quality assurance, human factors, and system security. Clause 5 also discusses the use of system prototyping and simulation in support of SEP activities, and identifies several different classes of analysis and assessment that could be applied during an iteration of the SEP (operational analysis, manufacturing analysis, disposal analysis, etc.).

2.1.5.2 Applicability to M&S Development

MIL-STD-499C seems to have has a limited utility as an SE framework for MSDBP. On the positive side, it is very thorough in describing all the SE activities needed to build just about any type of system, and the detailed characterization of every product produced during a SEP iteration should leave no doubt as to what is required of system developers. It also emphasizes iteration among activities both within and across life cycle stages, which is generally the way M&S tools are built in practice. It also correctly emphasizes the need to tailor the process to different needs, although tailoring seems to be limited to what control gates are used (e.g., technical reviews, product milestones) or the number of SEP iterations needed to produce the product.

The main limitation in using this standard to serve as the MSDBP SE Framework is the different intended user groups for these products. MIL-STD-499C is very DoD-centric (by design), and was written as a compliance document for contractors of DoD systems. Due to the wide variety of different systems produced by the DoD, there is a correspondingly long list of requirements for the types of activities performed by the contractors and the interim products they produce. The system life cycle employed in this document corresponds to the DoDI 5000.2 acquisition phases, as would be expected for its intended user group. Thus, it covers the full system life cycle, from early requirements development through product disposal.

The MSDBP SE Framework is specifically focused on defining best practices for stand-alone M&S tool development. Many of the users of this process will be commercial companies, who will have little interest in complying with a U.S. military standard. These users will not want a set of rigid requirements, but rather useful advice (in the form of best practices) for improving their internal M&S development processes. Many of these users are expected to be small nimble companies that cannot tolerate an overly heavyweight development process (with extensive documentation requirements and numerous technical and managerial reviews), and while most of the tasks identified in MIL-STD-499C are only indirectly applicable to M&S development, those few tasks that are specifically M&S-related say very little about the process of developing and testing the tools themselves.

In summary, the major takeaway from MIL-STD-499C seems to be the activities and products defined by the SEP, and the general strategy of iterating the SEP across the various life cycles. While the many requirements defined in this document cannot be used as actual hard requirements in the MSDBP document, the requirements do identify the lower-level tasks that need to be performed, and thus could serve as the basis for at least some best practices in the M&S domain.

2.1.6 INCOSE Handbook

2.1.6.1 Summary

The purpose of the INCOSE Handbook, as identified in the standard itself, is to "define the discipline and practice of systems engineering for student and practicing professional alike" [Reference (f)]. It was developed to be entirely consistent with ISO/IEC 15288, but provides a deeper level of description to the processes and activities needed to execute the generic ISO/IEC 15288 standard. In general, it is stated that the INCOSE Handbook can 'serve as a reference to practices and methods which have proven beneficial to the systems engineering community at large and which can add significant value in new domains if appropriately selected and applied."

The INCOSE Handbook, since it is largely based on the ISO/IEC 15288 standard, takes a similar position with respect to lifecycle processes. Thus, it uses the same lifecycle model and four core process groups (Agreement Processes, Organizational Project-Enabling Processes, Project Processes, and Technical Processes). Further, it introduces the "Vee" Model, which is used to conceptualize the progression of system engineering activities during the various lifecycle stages, with particular attention to the concept and development stages (Figure 2-8).

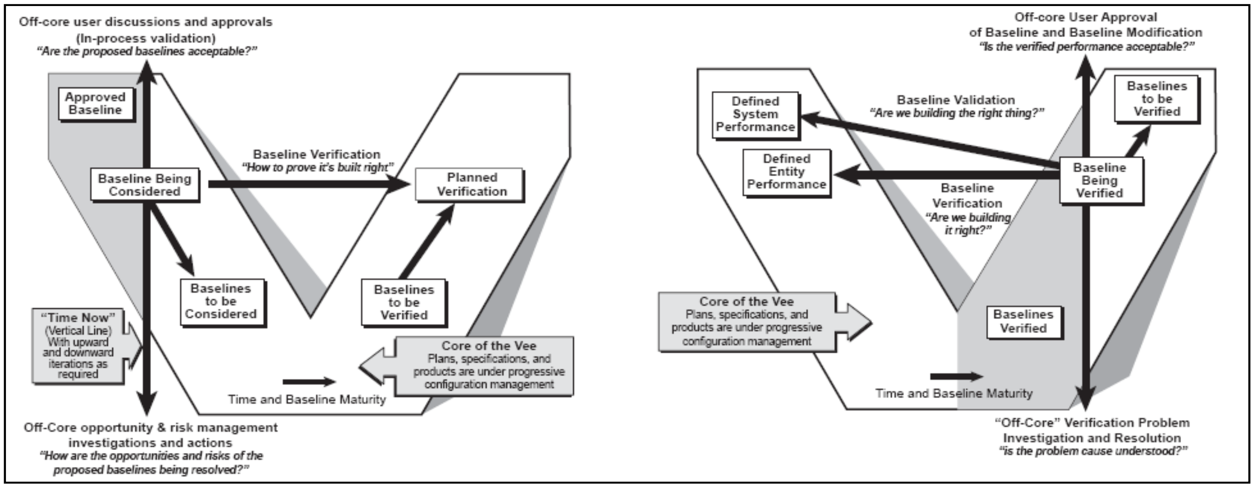

Figure 2-8: Left and Right Sides of Vee Model

The INCOSE Handbook also discusses the two fundamental approaches to system development. First, it describes what is referred to as "Plan-Driven Development", which uses the requirements/design/build/test/deploy paradigm to define a systematic approach that adheres to formalized processes as a system moves through a series of representations from requirements to design to finished product. Thus, there is considerable attention to the completeness of documentation, traceability from requirements, and verification of each representation. Then, "Incremental and Iterative Development" is described. This approach is used when requirements are unclear from the beginning or the customer wishes to hold the system-of-interest open to the possibility of inserting new technologies. Under this approach, a candidate system-of-interest is developed based on an initial set of assumptions, and then assessed to determine if it meets user needs or requirements. If not, another evolutionary round is initiated. This process is repeated until there is a satisfied user or resources are exhausted. In addition to introducing these different approaches, the Handbook also discusses how to choose among these approaches, along with supporting case studies. In general, M&S development would seem to be best supported by the Incremental and Iterative Development approach, due to the flexibility of implementation and the iterative nature by which M&S requirements are "discovered" on most projects.

The technical processes described in Section 4 of the Handbook are a mirror image of the technical processes described in the ISO/IEC 15288 standard. However, the organization of the process descriptions is somewhat different. In the 15288 standard, each process is characterized by a Purpose section, an Outcomes section, and an Activities and Tasks section. In the

Handbook, process descriptions also begin with a Purpose section, but then include sections for Inputs and Outputs rather than a single Outcomes section. In addition, the Inputs and Outputs sections tend to refer to explicit artifacts/products, while the Outcomes section provides a more generic description of a successful process implementation.

The Handbook also includes a Process Activities section, which corresponds to the Activities and Tasks section from the 15288 standard. However, there are some important differences. In the 15288 standard, each activity is defined according to a set of tasks, where nearly all tasks are accompanied by a NOTE that provides a deeper level of explanation and rationale for that task. In the Handbook, the Process Activities correspond to tasks in the 15288 standard, and while the Process Activities are generally consistent with the 15288 task descriptions, the exact wording used and the way the information is organized is quite different. The Process Activities also include a Common Approaches and Tips section that leverages much of the guidance provided in the task NOTEs from the 15288 standard.

The INCOSE Handbook includes a number of appendices that provide additional helpful information for implementing the technical processes. For instance, Appendix I provides useful "how-to" information on requirements identification, capture, analysis, and management. In addition, Appendix J provides user guidance on the functional analysis and allocation phase of development, and Appendix K describes the process of system architecture synthesis. All of these appendices are intended to be complementary to, and expand upon, the technical processes discussed in Section 3.

2.1.6.2 Applicability to M&S Development

Since the technical processes described in the Handbook mirror those in the 15288 standard, the conclusions for how well these processes could serve as the foundation for the MSDBP SE Framework are approximately the same. Thus, while most of the technical processes are indeed applicable to M&S, the same issues with respect to scope and activity sequencing are relevant for the Handbook as well. However, the Handbook does provide some additional benefits in some areas. For instance, the Handbook is by definition a guidebook, and is not subject to the rigidity concerns of a formal standard. Also, the Common Approaches and Tips associated with the Process Activity descriptions are a good source for the best practices themselves, as are the supporting appendices for the technical processes.

It is also important to note that the Handbook pays close attention to the other lifecycle processes that complement the technical processes. This is true in other SE process documents as well, but the description of the Project Processes, Enterprise and Agreement Processes, and Systems Engineering Support Activities (Sections 5, 6, and 8 respectively) have an especially strong emphasis on the many planning, control, and management activities that are needed for technical processes to be successful. While the MSDBP SE Framework is intended to be primarily technical in nature, it is apparent that there are so many potential best practices in the supporting lifecycle processes that some means of capturing these best practices must be included in the MSDBP SE Framework.

2.1.7 Capability Maturity Model Integration for Development

2.1.7.1 Summary

CMMI is a process improvement maturity model for the development of products and services. It consists of best practices that address development and maintenance activities that cover the product lifecycle from conception through delivery and maintenance. Prior designations of CMMI for systems engineering and software engineering (CMMI-SE/SW) were superseded by the current document “CMMI for Development” (Version 1.2) to reflect the comprehensive integration of these bodies of knowledge. The purpose of CMMI for Development is to provide a comprehensive integrated solution for development and maintenance activities applied to products and services.

A CMMI “constellation” is a collection of CMMI components that includes a model, its training materials, and appraisal-related documents for an area of interest. CMMI-DEV is the first of such constellations (note: other constellations are planned for services and acquisition).

CMMI-DEV defines a series of “process areas.” A process area is defined as a cluster of related best practices in a specific area, which when implemented collectively, satisfy a set of goals considered important for making significant improvement in that area. Each process area is further characterized by the general category that area supports. Table 2-2 summarizes this information.

Table 2-2: Process Areas and Categories

Process Area	Category
Causal Analysis and Resolution	Support
Configuration Management	Support
Decision Analysis and Resolution	Support
Integrated Project Management +IPPD	Project Management
Measurement and Analysis	Support
Organizational Innovation and Deployment	Process Management
Organizational Process Definition +IPPD	Process Management
Organizational Process Focus	Process Management
Organizational Process Performance	Process Management
Organizational Training	Process Management
Product Integration	Engineering
Project Monitoring and Control	Project Management
Project Planning	Project Management
Process and Product Quality Assurance	Support
Quantitative Project Management	Project Management
Requirements Development	Engineering
Requirements Management	Engineering
Risk Management	Project Management
Supplier Agreement Management	Project Management
Technical Solution	Engineering
Validation	Engineering
Verification	Engineering

The CMMI-DEV description of each process area includes the following information:

- Purpose
- Introductory Notes
- Related Process Areas
- General/Specific Goals
- General/Specific Practices
- Typical Work Products
- Subpractices

The CMMI-DEV concept of goals and practices is quite analogous to the concept of activities and tasks in IEEE 1516.3 and ISO/IEC 15288. That is, each goal in a process area represents an activity that must take place to implement some aspect of the process, and each practice is intended to provide user guidance as to the tasks that need to be performed to successfully complete that activity. Put another way, the goals define "what" needs to be done, and the practices define "how" to best achieve the goals. The typical work products and lower-level subpractices provide additional and a deeper level of user guidance for appraisal and improvement of existing user practices.

2.1.7.2 Applicability to M&S Development

In principle, the intended use of the CMMI-DEV is a very close match to what the MSDBP SE Framework is intended to provide. That is, it identifies a set of activity bins (i.e., goals) into which a corresponding set of best practices is defined. Although the MSDBP SE Framework is much more focused on basic user guidance rather than process appraisals, the use of best practices to guide users through the various processes inherent to systems development is generally what is intended for MSDBP.

One apparent deficiency with respect to the application of CMMI-DEV for MSDBP is the absence of the temporal dimension from the end-to-end process. That is, while each individual process area is very well defined, there is little to suggest exactly what the sequence of process area implementations should be, and how each process area relates to every other process area. Note that process area relationships are discussed in Section 4, but only within each individual process category. Also, although there is a short subsection on process recursion and iteration, there is little guidance regarding how the overarching development process would apply these considerations.

Another potential deficiency is the general coarseness of the major development phases. Since the MSDBP SE Framework is intended to reflect a technical process, the CMMI-DEV process areas of greatest interest are those in the Engineering category. Here, there are a total of six process areas defined:

- Requirements Development
- Requirements Management
- Technical Solution
- Product Integration
- Verification
- Validation

Requirements development and management are clearly important for stand-alone M&S development, and these are discussed at an appropriate level of description for what is needed for the MSDBP SE Framework. Similarly, verification and validation are both explicit process areas that are described at a level commensurate with MSDBP needs. However, this only leaves two process areas for every other aspect of stand-alone M&S development. Under "Technical Solution", there are only three activities/goals identified:

- Select Product Component Solutions
- Develop the Design
- Implement the Product Design

For the MSDBP SE Framework, the majority of the overarching process description is likely to be associated with this one CMMI-DEV process area. At a minimum, this implies that the three activities/goals for this process area would need to be broken out as separate process areas themselves, with (potentially) the underlying practices being elevated one level to activities/goals. There are also three activities/goals identified under the "Product Integration" process area:

- Prepare for Product Integration
- Ensure Interface Compatibility
- Assemble Product Components and Deliver the Product

While some of these activities/goals may be appropriate for compositional approaches to M&S development, the very fact that MSDBP is being explicitly targeted to stand-alone M&S development implies that integration activities are likely to be of somewhat lesser importance. Thus, inclusion of such integration activities in the MSDBP SE Framework would need to be heavily caveated.

2.2 SYSTEMS ENGINEERING FRAMEWORK ASSESSMENT

In order to assess the utility of the various SE processes in terms of their applicability to stand-alone M&S development, it is important to state a few up-front assumptions. First, the focus of the MSDBP project is to identify best practices across the technical activities necessary to build stand-alone M&S applications. All of the SE frameworks reviewed in this paper cover these technical considerations, but most do not treat the technical activities as any more or less important than other activities devoted to project management practices. While the importance of such issues is acknowledged, the amalgamation of technical and non-technical activities and tasks within the same SE Framework is likely to cause confusion to the targeted user base for the MSDBP product. Still, some issues with project management practices, such as risk management and configuration management, represent areas where best practices for M&S development are highly relevant. Thus, it is necessary to separate those activities/tasks focused on project management practices that are in-scope versus being out-of-scope, and determine how to include the in-scope project management practices considerations in the MSDBP SE

Framework while ensuring a clear, easy to understand representation of the core technical development process.

Of all of the SE frameworks reviewed in this paper, IEEE Standard 1516.3 is the only one that is specifically tailored for the M&S domain. Thus, it already includes all of the various activities and tasks that are necessary for building M&S environments. However, this standard is only relevant to distributed M&S development, and does not include certain activities that are relevant to stand-alone M&S development. Moreover, it is considered to be a mature, widely recognized standard in the M&S community and will be a helpful starting point for comparing what developers of stand-alone M&S will need along with the potential contributions of other SE frameworks. The best and most relevant aspects of all the SE frameworks assessed can be incorporated as appropriate, and elements of each of the SE frameworks that are either irrelevant to stand-alone M&S development or are better defined in competing frameworks can be extracted from the MSDBP framework.

The remainder of this section is focused on defining an appropriate SE framework for MSDBP. The methodology in this case is to evaluate the applicability of assessed SE framework activities and tasks for stand-alone M&S development, and as deficiencies are identified, evaluate which, if any, other SE frameworks define activities and tasks that are both applicable to the M&S domain and address the identified deficiencies. Those framework elements are then used as appropriate to support the anticipated needs of MSDBP users.

2.2.1 SE Framework Evaluation

Step 1*:* *Define Objectives:* The user, the sponsor, and the project development team define and agree on a set of objectives and document what must be accomplished to achieve those objectives.

Step 1 Activities*: Identify User/Sponsor Needs, Develop Objectives*

Step 1 Assessment*:* This early stage of the technical process involves having the sponsor of the activity define the model's intended uses, communicate that intent to the development team, and gain concurrence among the sponsor and development team as to the key objectives of the effort (including developmental constraints). Initial planning for the effort is also conducted at this time.

Each of the SE frameworks has "Requirements Development" as part of the technical process, as described in the previous section and in the side-by-side comparison. However, in the standard specific to the M&S domain, a distinction is made between an initial set of objectives and a more detailed set of requirements and is specified in a separate step. Such distinctions are made in other SE frameworks as well, but the specific terminology used is quite different and the step may be included in other technical processes, but not specifically called out as a separate step. The analogous terms used to identify the initial set of objectives contained in the various standards reviewed are as follows:

- IEEE 1516.3 – *Federation Objectives*
- ISO/IEC 15288 – *Stakeholder Requirements*
- MIL-STD-499C – *Functional Requirements*
- CMMI-DEV – *Customer Requirements*

The analogous terms used to identify the more detailed set of requirements contained in the various standards reviewed are as follows:

- IEEE 1516.3 – *Federation Requirements*
- ISO/IEC 15288 – *System Requirements*
- MIL-STD-499C – *Performance/Design Constraint Requirements*
- CMMI-DEV – *Product Requirements*

The SE framework standard specifically tailored for the M&S domain is unique in spreading the requirements development process across different developmental phases. Nearly all other SE frameworks describe requirements development/analysis as the initial phase of development, although several discuss the iterative nature by which requirements evolve (at least through design). Also, many SE frameworks discuss the importance of requirements validation.

From the perspective of stand-alone M&S development, there is no strong reason why the general topic of requirements cannot be fully covered in a single framework "bin". Here, the proposed approach is the use of the CMMI-DEV construct of a single overarching phase dedicated to requirements, using the same title (*Requirements Development*). At the activity level, it is proposed that the "Analyze" component of the CMMI-DEV *Analyze and Validate Requirements* activity be moved into the *Develop Product Requirements* activity, so that requirements validation stands alone as a separate activity (as is done in IEEE 1220, also known as ISO/IEC 26702 standard, and suggested by other frameworks). Also, it is proposed that the terminology used to describe the two different classes of requirements be drawn from the ISO/IEC 15288 standard and INCOSE Handbook, as the terminology seems most clear in these documents. Thus, to summarize:

Phase 1: *Requirements Development*

Activity 1: *Develop Stakeholder Requirements*

Activity 2: *Develop Product Requirements*

Activity 3: *Validate Requirements*

Step 2*: Perform Conceptual Analysis:* Based on the characteristics of the problem space, an appropriate representation of the real world domain is developed.

Step 2 Activities*: Develop Scenario, Develop Federation Conceptual Model, Develop Federation Requirements*

Step 2 Assessment*:* This need for conceptual analysis appears to be unique to the M&S domain and as such, the terms used in this step are based on the lexicon contained in IEEE 1516.3. For all other SE frameworks, some design activity always follows requirements development, presumably since [for most hardware (H/W) or software (S/W) systems preliminary system architecture can be derived directly from a validated set of requirements. In the case of either stand-alone or distributed M&S, there needs to be some validated representation of the real world missions/operations that are going to be captured in the M&S before M&S designers fully know what functions the M&S system must perform. Thus, this process of conceptual modeling needs to be addressed in the SE framework in some way.

In terms of the other IEEE 1516.3 activities in this step, nothing really needs to be captured in the MSDBP SE Framework. Obviously, the *Develop Federation Requirements* activity was addressed in the previous phase, and the *Develop Scenario* activity can be considered to be out-of-scope. In this latter case, the reason for this assertion is that the MSDBP SE Framework is meant to address the process of developing stand-alone M&S applications, and does not cover the execution aspects of that application. In this way, the Framework will be applicable to both government and commercial developers of M&S products, and not only those M&S developers that are building the M&S application as a means to some other end. Keeping the development aspects of the process separate from the execution aspects of M&S usage will allow the MSDBP SE Framework to address the largest number of potential users.

Removing these two activities from this step means that there is only a single activity associated with this step. This could suggest that conceptual analysis doesn't necessarily need to be identified as a unique phase of M&S development. In fact, since conceptual analysis is a way of conveying representational requirements to M&S developers, one possible approach is to include this activity in the *Requirements Development* phase (as part of *Develop System Requirements*). However, this masks the general importance of this activity, and would also result in best practices in this unique area being mixed with more general best practices for system requirements development. In addition, since requirements validation has been identified as a unique "bin" for identifying best practices, it can be easily argued that a corresponding activity is needed for conceptual modeling. Thus, this phase of development would have two activities, and not one.

It is important to note that, in the IEEE 1516.3 standard, the reason that conceptual modeling is included in the same step as *Develop Federation Requirements* is because of the close relationship among these activities. As the conceptual model evolves, the performance and behaviors of key entities becomes clearer, which helps to elucidate the system requirements. In addition, more detailed information on critical system characteristics helps to better define needed behaviors, which may affect the conceptual model. The fact that *Requirements Development* and *Conceptual Analysis* are reflected as separate phases in the MSDBP SE Framework should not be interpreted as meaning that one has to be completed before the other. Rather, these are simply containers for best practices, and the processes/activities associated with these containers can be implemented in many different ways (e.g., concurrently, iteratively).

Thus, the bottom line …

Phase 2: *Conceptual Analysis*

Activity 1: *Develop Conceptual Model*

Activity 2: *Validate Conceptual Model*

Step 3*: Design the M&S application or system:* Produce the design for the product or system, and developed a plan for system development and implementation.

Step 3 Activities*: Select Components, Prepare Design, Prepare Plan*

Step 3 Assessment*:* In the IEEE 1516.3 standard, the first step of this activity is relevant only to distributed M&S applications, so this activity will not be included in the MSDBP SE Framework. Clearly, the next activity (*Prepare Design*) is very applicable, although the nature of the design work to build stand-alone M&S applications is going to be different from that of distributed M&S applications.

Not surprisingly, all of the SE frameworks address system design as a critical component of the overarching systems engineering process, but in different ways. EIA-632 puts requirements definition under system design, and then just adds a single activity for *Solution Definition*. ISO/IEC 15288 and the INCOSE Handbook refer to architectural design rather than system design, although these two concepts are largely synonymous. They both include activities related to architecture definition, analysis, evaluation, documentation, and maintenance. IEEE 1220 focuses on the core six steps of *Requirements Analysis, Requirements Validation, Functional Analysis, Functional Verification, Synthesis,* and *Design Verification* inside what could be multiple *Preliminary Design* and *Detailed Design* loops. The *Functional Analysis/Allocation* and *Synthesis* phases described in MIL-STD-499C are roughly analogous to the notion of *Preliminary Design* and *Detailed Design* in IEEE 1220, in that both emphasize a functional decomposition of the system before allocating physical subsystems to functional components. System design is discussed under "*Technical Solution*" in CMMI-DEV, which includes design activities at both the component and full system level.

From the perspective of stand-alone M&S development, it is most important to capture the idea that an initial functional description of the M&S application is needed to validate that all requirements are being properly met, which is then followed by a detailed design consisting of physical[5] components and their interfaces. Both IEEE 1220 and MIL-STD-499C use the term "*Functional Analysis*", which works well for the desired SE Framework. Both of these documents also use the term "*Synthesis*" for the transformation from functional design to physical design, which will be adopted as well.

IEEE 1220 also identified *Design Verification* as an important activity in this part of the overall process. Since both Requirements Development and Conceptual Analysis identify activities for validation, it would seem to make sense to include a similar activity in the design phase. However, while IEEE 1220 identifies separate verification activities for each of the two major design activities, it is proposed that a single "bin" be included in this part of the SE Framework for all design verification tasks. Note that it should be very easy/straightforward for MSDBP users to determine whether best practices in this bin belong to the functional analysis or detailed design aspects of this developmental phase.

Finally, IEEE 1516.3 includes an activity in this phase for project planning. This is a continuation of planning activities that actually began in Step 1. While proper planning is critically important to the success of a project, planning is less of a technical concern than a project management issue that transcends all phases of development. For that reason, it is proposed that this be considered an in-scope management issue that will be included in the SE Framework but outside of the core technical process. This will be discussed later.

The bottom line …

Phase 3: *Product Design*

Activity 1: *Perform Functional Analysis*

Activity 2: *Synthesize Design*

Activity 3: *Verify Design*

Step 4*: Develop M&S application:* The M&S application is built as defined by the product design.

Step 4 Activities*: Develop the Application, Establish Agreements, Implement Designs, Implement Infrastructure*

Step 4 Assessment*:* The actual development of the M&S application is addressed differently in the various SE processes. In IEEE 1516.3, the activities begin to strongly reflect

[5] In the case of M&S, a "physical" component is meant to refer to an instantiated software component.

the nature of distributed M&S development. M&S application development, in this case, refers to defining the runtime data exchanges that needs to occur among distributed components, which is clearly irrelevant to stand-alone M&S development. Agreements in distributed M&S applications are also largely irrelevant, although the M&S development team may want to document any standards and/or conventions they intend to use during development. The implementation of distributed application infrastructure (e.g., bridges, gateways) is also largely irrelevant, although some infrastructure (e.g., operating systems, host computers) will be needed to ensure an adequate environment for software development. Thus, the key IEEE 1516.3 activity in this phase of development is the *Implement Federate (Component) Designs* activity.

In EIA-632, develop M&S application is called *Product Realization*, which includes both *Implementation* and *Transition to Use* activities. In ISO/IEC 15288 and the INCOSE Handbook, development is broken out across two major steps (*Implementation* and *Integration*). In IEEE 1220, this aspect of the process is referred to as *Fabrication, Assembly, Integration, and Test (FAIT)*. In CMMI-DEV, aspects of development are included in both the *Technical Solution* phase (*Implement the Product Design*) and the *Product Integration* phase (*Assemble Product Components and Deliver the Product*).

Due to the diversity of terminology across the SE frameworks, it is proposed that the generic term "*Product Development*" be used for the MSDBP SE Framework. This is consistent with the use of the term "*Product Design*" used in the preceding phase, and is broad enough to include any integration that may need to take place (e.g., for compositional development approaches).

Although most of the various SE frameworks provide useful guidance for product development, this guidance is more at the level of best practices than at the level of a specific development activity into which best practices can be placed. Thus, there are not many unique activities to include into this phase. Certainly, an activity called "*Implement Product Design*" will be needed to capture best practices in this area. However, the only other activity which is needed in this area is one to establish the development environment that will be used to create the desired M&S application. This could be a fully-integrated commercial software development environment, a set of off-the-shelf commercial software components that can be assembled into a suitable software development environment, a public domain set of components, or perhaps a software environment that was created by the M&S developer. In any of these cases, an SE Framework bin to capture best practices for how to acquire or build the supporting software development environment should be quite useful to future MSDBP users.

In summary,

Phase 4: *Product Development*

Activity 1: *Establish Software (S/W) Development Environment*

Activity 2: *Implement Product Design*

Step 5*: Plan, integrate, and test M&S application:* All necessary application integration activities are performed, and testing is conducted to ensure that interoperability requirements are being met.

Step 5 Activities*: Plan Execution, Integrate M&S application, Test M&S application*

Step 5 Assessment*:* In this step, the first of these activities will be addressed in a separate SE Framework element for the in-scope management issues, and the second concern (integration) was addressed in the previous step. Thus, testing is the main concern of interest in this step for stand-alone M&S developers. In several of the SE frameworks, testing is discussed within the context of product verification and validation activities. Thus, the product is tested to determine whether it correctly implements the stated design (verification) and tested to determine if the product performs/operates as intended (validation). Since this seems to be the convention in existing SE frameworks, it is proposed that this phase of development be referred to as "*Product Testing*," with two main activities, "*Perform Product Verification*" and "*Perform Product Validation*."

Phase 5: *Product Testing*

Activity 1: *Perform Product Verification*

Activity 2: *Perform Product Validation*

Note that many SE frameworks extend the technical processes to include issues such as transition, execution, analysis of results, maintenance, and disposal. These activities are about the application of the M&S product, not about its development. Since companies that build stand-alone M&S applications for profit (or not-for-profit, but for general use) are considered to be part of the MSDBP user base, one cannot assume that the M&S application is only being developed to address a particular analytic problem (i.e., a means to some other end). Thus, the MSDBP SE Framework is being designed to only address the development aspects of stand-alone M&S applications, and not the execution aspects. For this reason, none of the post-testing activities addressed in the reference documents are applicable to MSDBP. Therefore, since the intent of the MSDBP SE Framework is to focus on developmental best practices, it is assumed that such longer-term considerations are out-of-scope. As a result, the last phase of the technical process for the development of stand-alone M&S applications is assumed to be final product verification/validation.

2.2.2 Considerations for Project Management Practices

As was discussed earlier, there are many systems engineering activities that support the technical processes. All of the various SE processes discuss issues related to project management practices to varying degrees. ISO/IEC 15288 and the INCOSE Handbook provide the richest set of supporting considerations for project management practices. To properly account for these concerns in the MSDBP SE Framework it is proposed that a single category, equivalent to one of the five phases described above, be incorporated into the Framework (entitled "*Supporting Project Management Practices*"). However, creating separate activities for every potential concern with project management practices would result in far too many disparate bins for the best practices. Instead, it is proposed that a more aggregated set of activities that focus on project management practices be derived from ISO/IEC 15288 and the INCOSE Handbook based on their perceived relevance to stand-alone M&S development. The five specific activities for project management practices that are proposed for this SE Framework category include:

- Project Planning
- Project Control/Resource Management
- Risk Management
- Quality Management
- Configuration Management

2.2.3 Side-by-Side Comparison

Appendix C synthesizes a comparison of the SE processes analyzed in this section.

Section 3 proposes a framework that integrates the five proposed technical processes with this set of project management practices.

2.3 LITERATURE SEARCH

A literature search was a key component of the study team's practice discovery process. The team searched the most recent five years' worth of proceedings from:

- Simulation Interoperability Workshops (SIWs)
- Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC)
- National Defense Industrial Association (NDIA) Systems Engineering Conference

The team also searched for other literature sources that might provide sound practices. A complete list of the additional sources identified is provided in Appendix D.

2.4 COMMUNITY SURVEY

2.4.1 Initial Participation Request

The initial approach for identifying sound practices was to identify organizations that develop models and simulations, and have established engineering practices for doing so. A wide net was cast, sending out an initial participation request to e-mail lists for SISO, NDIA M&S Committee, and the AMSWG. Team members also visited the booths of all the software developers at I/ITSEC 2008 and asked directly for their participation. A web page was created at JHU/APL to collect the responses. The questions posed in the initial request are below.

1. Does your organization develop models/simulations, supporting environments for developing models/simulations, or both?
2. Are your organization's practices based on industry standards or internally developed? [Industry standards – skip to Question 4]
3. Is your organization willing to provide a detailed description of these practices to the JHU/APL Study Team, assuming any intellectual property is properly protected by a non-disclosure agreement? [Internally-developed practices stop here]
4. Please name and provide appropriate references for the industry standards upon which your practices are based.
5. Please describe your tailoring of the industry standards for application within the M&S domain. If you would prefer to discuss this with the study team under a non-disclosure agreement to protect your intellectual property, please so indicate.

Forty-seven responses to the initial request were received, four of which indicated proprietary practices that would require an NDA to discuss. However, since the results of this effort are intended to be public, the team decided not to pursue these four responses further.

Most respondents develop both models/simulations and supporting environments. The respondents were almost evenly split between using industry standards and internally developed practices. The responses indicated that there was some confusion on the question about industry standards used because several responded with HLA and DIS rather than software or systems engineering processes, indicating a need to provide clarification in follow-on conversations. Surprisingly, fewer than half of respondents answered this question at all. Of those that responded, the industry standards preferred were CMMI (7), ISO 9000/9001 (5), INCOSE (1), and EIA-632 (1). As will be shown, this response influenced the team's choices of sources for the development of a systems engineering framework for integrating the best practices that were identified.

2.4.2 Detailed Sound Practice Request

Based on the results of the initial survey, especially Question 3, the team began soliciting more detailed responses from the initial respondents. The goal of this second round of questions was to elicit specific practices for integration into the SE framework. Because more detailed information was being sought, in this round, team members communicated with respondents in person, and via telephone conference and e-mail. The questions from this second round are below.

1. What are the specific activities that comprise your development process? Please provide a short description of each activity, and if applicable, how those activities align with your major developmental phases.
2. Which activities in your process do you consider to be the most critical to success, and why? For each of these activities:
   a. Please describe the lower-level tasks associated with these activities.
   b. Why are these tasks necessary?
   c. What are the underlying development practices associated with this activity that most contribute to the likelihood of success, and why?
3. Were all of your models, simulations, tools and/or frameworks developed using your best practices, or did the best practices evolve during the course of tool development, i.e., is process improvement part of the overall process?
4. What drove you to select/develop these practices for developing M&S tools? How long have you been using these practices?
5. How is the development of M&S tools different than the development of other software tools? How do the processes for their development compare?
6. If you were creating a standard for M&S development best practices, what would you include or not include in it and why?

Unfortunately, only a few responses to this second round of questions were received, probably due to survey fatigue. Additionally, a few respondents concluded that they did not feel comfortable sharing their processes publicly. Fortunately, the team never intended to depend solely on the surveys.

2.5 RESULTS OF PRACTICE IDENTIFICATION

The literature search and community survey netted approximately 116 practices. To enable the task of analyzing the practices, the team created a template, illustrated in Table 2-3, for capturing sound practices from all sources. Each practice was given a unique ID for later management, as well as a short descriptive title. If the practice came from the survey responses, the point of contact was included in case there was a need to go back to the source for additional information or clarification.

Table 2-3: Practice Template

ID #	**Short Descriptive Title**
SE Framework Category	**POC: Name, Email Address, Phone #; "literature" for literature search**
[Requirements Development, Conceptual Analysis, Product Design, Product Development, Product Testing, Project Management Processes]	
Description	
Rationale (Why the practice is useful/needed.)	
Source Reference: If derived from an industry standard, provide document name and version, and section number(s)	
Notes	
If this practice is derived from another source, complete the sections below.	
Rationale for Tailoring	
Description of Tailoring for M&S	

The most important field in the template is the description. It contains the substance of the practice. Many of the fields in the template are optional, but not this one. In some cases, the team identified additional information explaining why the practice is useful or needed; this information goes in the rationale field.

The source reference field is self-explanatory. Also included was a notes field as a catch-all for any additional information that might be useful in the later sorting and integration. Finally, since the team allowed for practices derived from industry standards and other sources, fields for capturing the specifics and rationale for tailoring from the original source were included.

2.5.1 Sound Practices Database

Only a handful of sound practices were obtained before the team realized that collecting them in individual files made the collection process easy, but it would make sorting and analyzing them difficult. The contents of all the completed templates were copied into a database to enable sorting, grouping, and analyzing sound practices. With the database, team members could search and sort by engineering phase, title, or source.

2.5.2 Observations on Collecting Sound Practices

The team faced several hurdles identifying sound practices specific to stand-alone M&S. First, there is little incentive for recording such practices. Papers about the models and simulations themselves and/or the useful results they produce are more interesting to write and read than papers about the process by which they were developed. Second, as was discovered with Question 5 of the detailed survey, most of the process of engineering models and simulations is the same as it is for engineering any good software. Of course, there was the issue of proprietary processes and practices. Sharing an organizations sound practices for engineering good models and simulations can be seen as revealing a market differentiator, especially for commercial organizations. Finally, the team encountered quite a bit of non-specific guidance in the literature. This guidance usually took the form, "when addressing issue A, consider B, C, D, and E." Unfortunately, in many cases there was no guidance on how to weight consideration of B – E, or on how to apply the results of such a weighted consideration. It is hoped that the publication of this best practices guide will energize this discussion and bring out more practices that can be incorporated in a subsequent version of the guide, perhaps even an open standard as will be discussed in Section 4.

2.6 BEST PRACTICE SELECTION CRITERIA AND ASSESSMENT

With the SE Framework and sound practices in hand, the team needed to determine which practices qualified as "best" practices, and to mediate overlaps and conflicts between similar practices. To make this process as objective as possible, the team developed a set of criteria for inclusion in the final guide.

The team took a three-step approach to developing the criteria. First a Google search on the words "best practices" was completed. This search resulted in a collection of practices across industries, domains, products, and languages, for example:

- COSO – Committee of Sponsoring Organization – Organizational Guidance <http://www.coso.org>
- Best Practices in Education <http://www.mlms.logan.k12.ut.us/wested_docs/west_research.htm>
- Best Practices in Technology Integration <http://www.remc11.k12.mi.us/bstpract>
- Best Practices in Web Page Construction <http://drupal.org/best-practices>
- Best Practices in Project Management <http://www.amd.com/us-en/assets/content_type/DownloadableAssets/Giga_-_Project_Management_Best_Practices-_Key_Processes_and_Common_Sense_(1-03).pdf>

Within the results, team members looked for common themes, terms, and concepts across entries, filtering the ones that were not relevant to M&S. For example, in "Relationship between Mobile Web Best Practices (MWBP) and Web Content Accessibility Guidelines (WCAG)" [Reference (k)], best practices for web accessibility are discussed that are not particularly relevant to M&S. While the subject of the best practices is technical, there is not a lot to be drawn because the practices specifically deal with attributes of accessibility and how to enable it.

In the second step, team members searched with narrower criteria, "software system engineering best practices" and "best practices modeling and simulation," and looked for specific terms and concepts. Finally, all the perspectives and filtered redundancies were combined, resulting in the list of criteria below:

- Specificity – Does the practice have demonstrated effectiveness within specific M&S domains?
- Comparability – Has the practice been compared positively to other practices in controlled studies (or could it be)?
- Degree of Independence – Is the practice platform or implementation independent?
- Efficacy – Does the practice support effective use of resources including intellectual capital?
- Customization – Does the practice allow customization and tailoring to an organization or domain's needs?
- Coherence – Does the practice align with other adopted best practices?
- Robustness - Does the practice usually result in a better outcome?
- Cohesion - Does the practice describe a single idea, concept or construct and not multiple ones grouped into a single practice?
- Coupling - Is the practice's adoption independent of other practices, i.e. does the adoption of this practice necessitate the adoption of another?
- Sustainability – Is it cost effective to sustain the practice after adoption?
- Usability – Can the practice be used, learned and employed in practice?
- Scalability – Is the practice scalable to projects of different sizes?
- Agility – Can the practice adapt readily to changing conditions, e.g., organization changes, contextual changes, etc.)?
- Generality – Is the practice expressed as generally as possible?
- Legal aspects – Is adoption of the practice free of difficult legal/proprietary aspects?
- Consensus – Is there widespread community acceptance of the practice?
- Cost Elasticity – Do the benefits of the results outweigh the cost of adoption of the practice?
- Repeatability – Does the practice repeatedly give desired results?
- Durability – Does the practice remain effective over time?
- Applicability – Is the technology related to the practice widely applicable and not just to a subset of problems or domains?

This process forms a core of the criteria, but the set was augmented to include some others that were empirically observed by the authors of the study.

As sound practices were identified and documented, team members tagged them according to the SE phases as described in Section 2.2. Once all the sound practices were identified, the study team reviewed the practices in each category, filtering them according to the evaluation criteria above, resulting in a set of "best" practices. Where there were conflicts/overlaps between closely related best practices, the additional rules stated above were applied, merging conflicts/overlaps into a single practice where appropriate.

The study team started with 116 practices in the database. First, approximately 10 practices that simply restated concepts already in the SE Framework, such as "develop a conceptual model," were removed. Then each team member performed an individual assessment:

- Which evaluation criteria the practice meets
- Where it belongs in the SE Framework (phase and activity)
- Whether the practice is M&S-specific[6]

Armed with the individual assessments, the team began the painstaking task of debating individual positions and reaching consensus about whether a practice qualified as "best" by passing multiple criteria, was sufficiently M&S-specific, and the SE Framework phase and activity to which to assign it.

Along the way, team members identified the need to clean up several practices. In some cases there were simple transcription errors from the original sources. As expected, the team also saw overlaps between practices from different sources that were merged into single practices. The study team also found instances where the rationale was merged with the practice, so the rationale was removed. Slight rewrites were performed where a sound practice was found that could be tailored to be M&S-specific, even though it wasn't initially stated as such. The final result of this effort was a set of 50 "best" practices that are integrated into the SE Framework in Section 3 and described in detail in Appendix B.

2.7 VETTING THE RESULTS

The agreement with the M&S community maintained that individuals and organizations that contributed to the sound practices would be allowed to review the results before they were shared with the broader community. This review process resulted in very few modifications and those modifications were very limited in scope.

[6] The team received a few practices that clearly qualified as "best" from a general systems or software engineering perspective, but weren't M&S-specific, e.g., a detailed configuration management process.

2.8 STUDY PAPERS AND PRESENTATIONS

During the course of the study, the study team had the opportunity to present several briefings on study progress as listed in Table 2-4.

Table 2-4: Study Papers and Presentations

Forum	Dates
AMSWG	12 August 2009
NDIA M&S Committee Meeting	13 August 2009
NDIA Systems Engineering Conference	29 October 2009
INCOSE Model-Based Systems Engineering Workshop	6 February 2010
2010 Spring SIW[7]	12 - 16 April 2010

2.9 SISO STUDY GROUP

To solicit feedback on the study, the study team formed a study group (SG) within SISO to provide input and feedback on the study at several stages. The planned tasks, as detailed in the SG terms of reference (ToR), were defined to be:

1. Review the sound model and simulation development practices identified by the JHU/APL team and provide additional inputs.
2. Review, and provide recommendations regarding, the systems engineering process chosen by the JHU/APL team as a framework for integrating the identified sound processes and defining one or more model and simulation development best practices.
3. Provide input to use cases for the application of classes of software best practices.
4. Interact with the JHU/APL study team regarding a plan for synthesizing best practices from the above sound practices within the above framework. This plan is expected to address the principles and process for doing so.
5. Participate in the review and refinement of the draft best practice(s) developed by the JHU/APL study team.
6. Determine feasibility of a SISO guidance product based on the final best practice(s) identified in the DoD study.

In addition to being a potential source of additional information beyond the surveys and literature search, the study group is a necessary first step in the SISO process if the results of the study are to form the basis of a SISO standard as suggested by Task 6 in the ToR. A kickoff meeting of the study group was held at the 2009 Spring SIW in San Diego, CA.

[7] The associated paper was nominated for a SIWzie, a best paper award.

One of the lessons learned from this task is that establishing such a study group (SG) and coordinating with it throughout a DoD study presents a schedule challenge because all materials to be shared with the SG require public release, and the volunteer nature of the SG makes it nearly impossible to get feedback in the timely fashion necessary for an established project schedule.

The study group will remain active beyond the end of this study for the purpose of determining the feasibility of developing a standard and reporting out to SISO.

This page intentionally left blank.

3 SYSTEMS ENGINEERING FRAMEWORK

In this section, a series of proposals in Section 2.2 is integrated into a single view of the framework, along with graphical and textual descriptions of different aspects of the framework. A graphical depiction of the MSDBP SE Framework is provided in Figure 3-1. There is a total of six major process elements in the framework; the five technical phases shown in the Technical Processes area, along with a sixth "umbrella" element for capturing best practices in each of the five areas of project management practices shown. Each of the technical processes is broken down into a set of lower-level activities, which will be the level at which best practices will be binned. Because practices under each activity might not have a chronological relationship to each other or might be performed in parallel, the practices have been binned under activities, but not ordered within the activities.

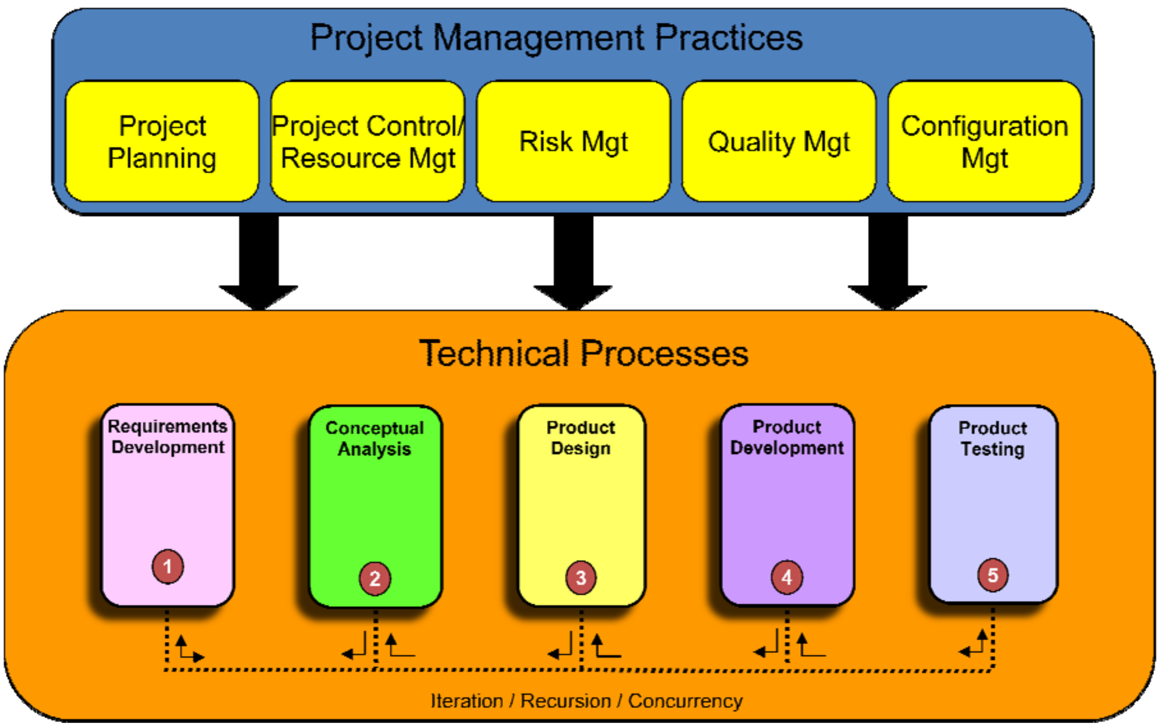

Figure 3-1: MSDBP SE Framework

The following is a textual description of all processes and activities within the framework. Together with the graphical representations, this will provide the organizational structure necessary for M&S developers to easily access and leverage the best practices most relevant to their efforts. Each activity is followed by a bulleted list of the titles of best practices identified for that activity. The full definitions of the practices are contained in Appendix B.

Requirements Development: The purpose of this phase of the M&S development process is to produce the set of requirements that will drive M&S design activities and provide the criteria by which the success of the M&S development project will be judged. This includes all categories of requirements, and all activities needed to ensure completeness and consistency of the requirements throughout the product lifecycle. Although requirements may be refined during any stage of the development process, all M&S best practices related to any aspect of requirements development, analysis, or validation will be captured in this section.

Develop Stakeholder Requirements: The purpose of this activity is to translate high-level stakeholder needs, constraints, intended uses, and expectations into an initial set of requirements for the M&S application. The term "stakeholders" in this case can mean anyone with an interest in influencing the nature of the M&S product, including sponsors, end users, and governing agencies. This activity includes defining the environment and constraints under which the product will be employed (the user space). This activity captures high-level requirements and adjudicates among competing interests to define what capabilities the M&S product must support and those that are merely desirable.

- Establish intent for model use.
- Use focus groups in simulation creation.

Develop Product Requirements: The purpose of this activity is to acquire necessary domain knowledge and translate the operationally-oriented stakeholder requirements into a more detailed set of system-level requirements as the basis for M&S design. These requirements should be directly testable, and thus provide the criteria by which the success of the development effort is measured. A proper analysis of the requirements should be conducted as part of this activity to ensure that the effort is feasible given stakeholder constraints (e.g., funding, facilities, and personnel), and that all stated requirements are both necessary and sufficient.

- Specify data content.

Validate Requirements: The purpose of this activity is to increase stakeholder confidence that meeting the stated requirements will result in an M&S application that fully meets their needs. This primarily requires validating the traceability between operational and system requirements, including validating that all stakeholder constraints are properly accounted for.

- Use survey methods to elicit Subject Matter Experts' (SMEs') knowledge.

Conceptual Analysis: The purpose of this phase of the M&S development process is to produce an implementation-independent conceptual depiction of the real-world missions and operations that must be represented in the desired M&S application. The product resulting from this activity is generally referred to as a *conceptual model*. This model can be used as the structural basis for many design and development activities and can highlight correctable problems early in the development of the M&S application when properly validated by the appropriate stakeholders.

Develop Conceptual Model: A conceptual model is developed to identify those relevant entities within the M&S domain of interest that should be represented in the product to satisfy validated requirements. The conceptual model identifies the attributes of each entity that should be represented, the behaviors/performance of each, and the static and dynamic relationships among the entities. Algorithms should be specified where feasible.

- Establish model focus by carefully choosing model behavioral aspects and data.
- Use a formal language for linking requirements to a conceptual model.
- Use a standard process for creating a conceptual model.
- Select computer scientists with domain expertise to be on the conceptual modeling team.
- Augment logical data models with semantics.
- Create a data dictionary.
- Include full simulation specification and context in a conceptual model for a simulation system.
- Format the conceptual model using a standard notation accessible to all stakeholders.
- Combine conceptual modeling with knowledge acquisition/knowledge engineering.

Validate Conceptual Model: The purpose of this activity is to validate that the conceptual model is accurate and complete with respect to the problem space of interest. This normally involves a review by SMEs to ensure that the defined representations are operationally correct, and that collectively, the conceptual model fully addresses all stakeholder requirements. Validation of the conceptual model is frequently done in parallel with requirements validation, due to the close relationship between these products.

- Document a rationale for realistic output measures.
- Use a standardized conceptual model to mitigate stakeholder subjectivity in simulation design.
- Involve the decision-maker in the model development process.

Product Design: The purpose of this phase of the M&S development process is to produce the design of the M&S application. This is normally conducted in an iterative fashion, with multiple loops of analysis, synthesis, and verification. The number of design loops is primarily driven by the size and complexity of the M&S application, as dictated by the system requirements.

Perform Functional Analysis: The purpose of this activity is to translate the validated system requirements and conceptual model into a complete set of required M&S functions. The functional architecture should include an organization (arrangement and sequence) of these functions and sub-functions that are necessary to properly reflect required system behaviors. Functional characteristics, such as entity states, trigger conditions, and functional interfaces (e.g., inputs/outputs) should all be captured at this time.

Synthesize Design: Using the functional analysis and conceptual model, the physical architecture of the product is synthesized. The architecture includes an arrangement of system elements, the decomposition of those elements (as required), internal and external interfaces, and design constraints. The final design solution should provide all of the information needed to support development of the M&S application, including:

- Maintain a distinction between models and simulations.
- Use archived models and model components from an authoritative source.
- Select input data items based on a complete problem context.
- Define uncertainty models.
- Use design patterns in M&S.
- Balance modeling needs with data considerations.
- Design data storage and retrieval architecture.
- Consider availability of data sources when designing simulation.
- Group and separate data from models with varying resolution.
- Distinguish unknowns from unknowables.
- Use intelligent analytical approaches to handle unavailable or unknowable data.
- Adopt commonly accepted icons, symbols, shapes, and colors used to represent simulation entities, where possible.
- Evaluate a model's pedigree before (re)using it as a component.
- Create both an analysis data model and a logging data model to facilitate capture and use of simulation output data.
- Use standards where applicable.
- Separate data I/O interface from model code.
- Use a standardized logical data model and format for I/O data.
- Select output data items based on a complete view of simulation usage.
- Design models as components with loose coupling.

Verify Design: The purpose of this activity is to verify that the system fully satisfies all stated requirements and is consistent with its conceptual model, both its functionality and its physical architecture. This is a continuous process during design, where traceability from requirements to functional representation to physical representation is the primary concern.

Product Development: The purpose of this phase of the M&S development process is to build the M&S application defined by the product design. This primarily involves implementing a controlled software development process to implement the product design, although even stand-alone M&S applications could potentially have hardware-in-the-loop. This phase generally requires considerable iteration with the testing activities defined in the subsequent phase.

Establish Software Development Environment: The purpose of this activity is to acquire and install (as necessary) an integrated set of supporting software and hardware assets that collectively establishes the environment needed for software developers to efficiently and effectively produce the desired M&S application. This includes such assets as operating systems, host computers, modeling frameworks (e.g., MATLAB), reusable code elements, debugging aids, and visualization equipment. A well-integrated development environment can significantly reduce the time and effort required to implement the M&S design and should include

- Use scenario generation tools to promote consistency and efficiency.
- Choose the right architecting tool for static and dynamic aspects of the M&S application.

Implement Product Design: The purpose of this activity is to produce the desired M&S application via implementation of the product design. This involves creating the software elements that correspond to design elements, and composing those software elements into a unified M&S system. The specific approach to implementation is at the discretion of the development team, subject to stakeholder constraints, but should

- Employ common random numbers in models.

Product Testing: The purpose of this phase of the M&S development process is to ensure that the developed M&S application meets all requirements and satisfies all stakeholder expectations. The output of this phase is the final product of the M&S development effort.

Perform Product Verification: The purpose of this activity is to confirm that the requirements are correctly implemented by the M&S application. This involves unit testing of individual M&S components, and system-level testing to ensure that all interfaces are working properly. As faults/errors are discovered, appropriate corrective actions must be defined and implemented, which may involve loopbacks to early design/development activities.

Perform Product Validation: The purpose of this activity is to ensure that the M&S application, when exercised for its intended use, will meet all of the operational needs articulated in the original stakeholder requirements. This may involve capability demonstrations or other such customer reviews, or could involve a trial-use period, whereby users report any faults/errors detected during normal use, for which corrective actions are defined and taken. Stakeholder satisfaction is the overriding goal of this activity.

- Collect referent information.
- Decompose qualitative SME input into quantitative indicators.
- Validate models against each intended use.

Project Management Practices: In order to conduct a successful project, there are many more considerations that must be effectively addressed in addition to the technical processes. Supporting project management practices are primarily management activities that overlay every aspect of the product development. While project managers are generally responsible for the conduct of such activities, M&S developers are full participants in ensuring that these activities are effectively assimilated into the normal day-to-day process of M&S development.

Project Planning: This activity produces and implements the planning resources necessary to assess and control the progress of the project. This includes the development of appropriate Work Breakdown Structures (WBSs) at the task level, including schedule/budgetary information and required personnel.[8]

Project Control/Resource Management: This activity involves the implementation of the project plan, which project managers use to continuously compare and assess progress toward project goals. This also involves ensuring that the use of available financial resources is within defined spending curves. Project control also involves managing the resolution of unanticipated problems/issues, and defining/implementing whatever corrective actions are necessary when deviations from the project plan are necessary.

Risk Management: This activity is concerned with how the various forms of risk (technical, schedule, cost, and programmatic) are identified and minimized throughout the M&S development process. Risk mitigation is concerned with reducing the consequences of undesirable developmental events and/or reducing the probability that such events will occur. An effective risk management plan will include approaches and techniques for identifying, analyzing, and treating whatever risks may be inherent in the chosen development strategy.

- Conduct engineering integration reviews.

Quality Management: This activity involves the management of whatever internal Quality Assurance (QA) policies and procedures overlay the development process to ensure that the final product meets defined quality standards. This generally involves conducting periodic

[8] The study team found no best practices specific to M&S in their research. However, Project Planning and Project Control are such general activities that they are usually the same regardless of whether the systems under development are hardware or software, much less a specific type of software.

reviews to assess the effectiveness, strengths, and weaknesses of the QA processes, including its applicability and application to the project at hand. The verification and validation activities described in the technical processes are a critical aspect of the broader QA effort.

- Include user domain representatives and external developers in peer reviews.
- Use SMEs throughout the development life cycle.
- Use Systems Engineering analysis and documentation.

Configuration Management: This activity involves the management of the evolving configuration of the M&S application, including requirements, documentation, data, and other artifacts, throughout its development. This normally involves the assignment of unique identifiers for each element or component of the M&S application, establishing controls so that component changes result in a consistent version of the application, and recording/tracking all product changes to maintain comprehensive traceability.

- Use a standardized method of "packaging" for developing model components.
- Document model abstraction decisions.
- Keep data current.
- Establish a configuration management system.
- Document models and simulation data with metadata.

3.1 SOFTWARE ENGINEERING PRACTICES

While team members are primarily concerned with systems engineering processes and practices, at some point the system design must be translated into software. At this point, the SE Framework intersects with software engineering practices. Not wanting to replicate all the previous good work on software engineering best practices, the team has produced a mapping of the SE Framework to various recognized software engineering processes. There are two basic categories of software engineering processes: linear and incremental/iterative. Two popular processes within each category were selected, as well as a “generic” process from a software engineering textbook. Mapping to linear processes is fairly straightforward. Mapping to iterative processes, on the other hand, is a bit more difficult, because iterative processes are “two-dimensional,” in that they have a macro-level view, e.g., phases, spirals, increments, that repeat more detailed development activities with a different focus each time. The SE Framework activities map to iterative processes in different ways. For example, “Develop Stakeholder Requirements” maps to the entire Spiral 1 of the Spiral process. “Validate Requirements” maps only to one particular section of Spiral 2. “Project Planning” maps to one particular section of **all** spirals. The following subsections summarize the individual software engineering processes. The side-by-side mapping to the SE Framework follows the individual summaries.

3.1.1 Generic Model

Table 3-1 summarizes the activities of the generic software engineering model.

Table 3-1: Generic Model Summary

Activity	Description
Communication	Understand stakeholders' objectives for the project and gather requirements that help define software features and functions.
Planning	Define software engineering work by describing technical tasks to be conducted, likely risks, required resources, work products to be produced, & work schedule.
Modeling	Create models to understand software requirements and the design that will achieve those requirements.
Construction	Generate code and test.
Deployment	Deliver software product to customer, who provides evaluation and feedback.

3.1.2 Waterfall Model

Table 3-2 summarizes the activities of the waterfall software engineering model [Reference (l)].

Table 3-2: Waterfall Model Summary

Activity	Description
System Requirements	Develop overall requirements for the system.
Software Requirements	Develop requirements for the software component of the system.
Analysis	Analyze requirements.
Program Design	Design the program.
Coding	Generate code.
Testing	Test software.
Operations	Deploy, use, and update the software.

3.1.3 V-Model

The V-model [Reference (m)] is an adaptation of the waterfall model emphasizing the relationship between testing and other phases of the development cycle. The model has four sub-models, of which only System Development is broken out here, as it is most closely related to the SE framework. The other three sub-models are Project Management, Quality Assurance, and Configuration Management which all map to the Project Management Practices. Table 3-3 summarizes the activities of the V-model software engineering model.

Table 3-3: V-Model Summary

Activity	Description
Requirements Analysis	Elicit stakeholder requirements.
System Design	Determine how requirements can be implemented.
Architecture Design	Create high-level design.
Module Design	Create low-level design.
Coding	Implement design.
Unit Testing	White-box testing.
Integration Testing	Black-box testing to expose interface faults between components.
System Testing	Test against system specifications.
Acceptance Testing	Test against system requirements.

3.1.4 Spiral Model

The spiral model [Reference (n)] is an incremental model that repeats activities (below, Table 3-4) in spirals (below, Table 3-4) as illustrated in Figure 3-2 [Reference (o)]. Output products differ between phases, though the general activities are the same. Table 3-4 summarizes the activities of the spiral software engineering model.

Table 3-4: Spiral Model Summary

Activity	Description	Spiral (Iteration)	Description
Determine Objectives	Understand stakeholders' objectives for the upcoming iteration, including alternatives and constraints.	0: Feasibility Study	Determine feasibility of proposed system.
Identify and Resolve Risks	Conduct risk analysis for the iteration, begin/continue prototype development.	1: Concept of Operations	Create, evaluate, and test concept of operations for the system.
Develop and Test	Carry out planned work, developing and testing the work product.	2: Top-level Requirements Specification	Derive requirements for the system.
Plan Next Phase	Evaluate completed work product, integrate and test as needed. Plan next iteration: define upcoming work by describing technical tasks to be conducted, required resources, work products to be produced, and work schedule.	3: Design	Design the system.
		4: Implementation	Implement and test the design.

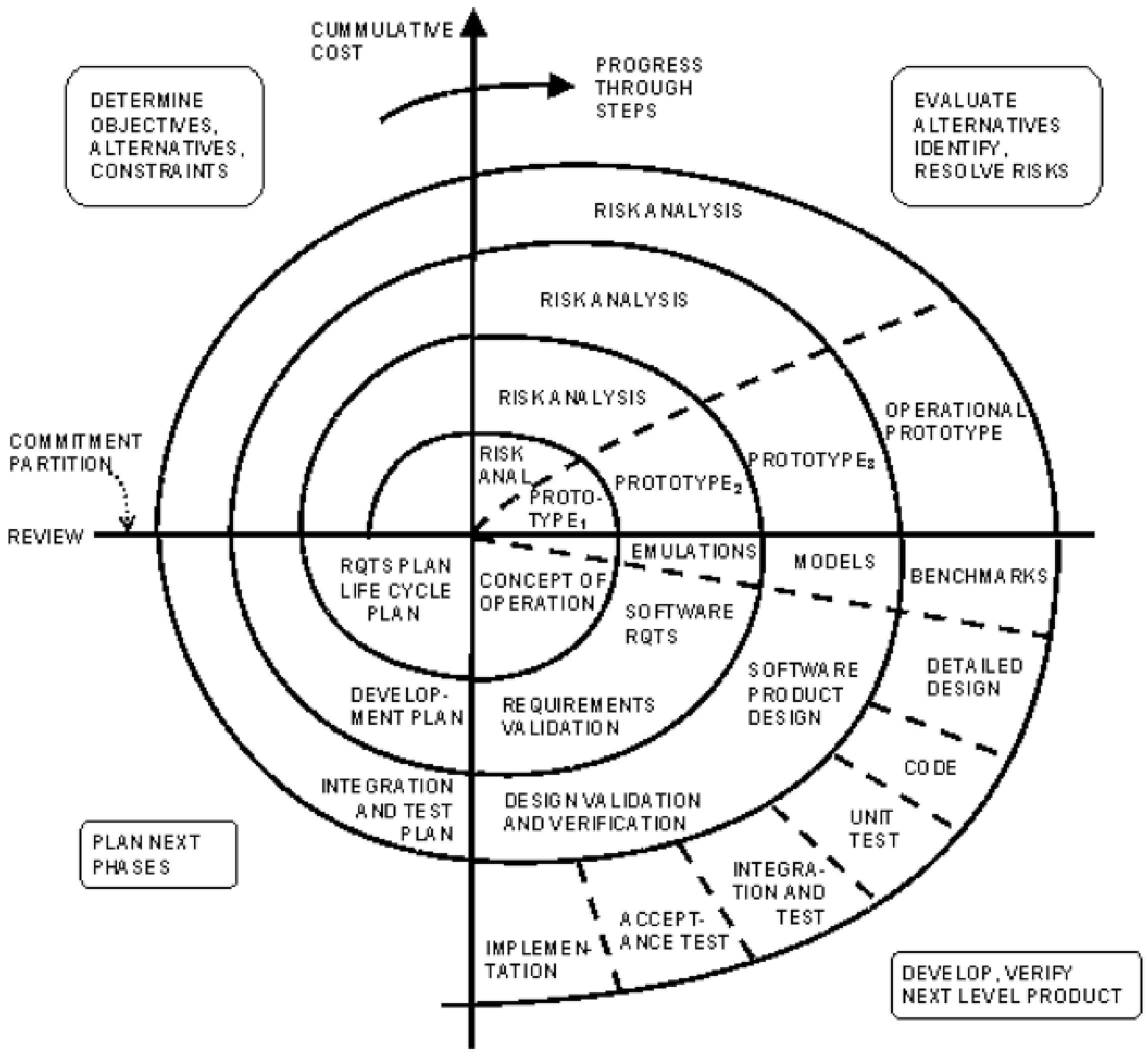

Figure 3-2: Spiral Model

3.1.5 Rational Unified Process

The Rational Unified Process (RUP) [Reference (p)] is an incremental process composed of four lifecycle phases that are spanned by various engineering and supporting "disciplines." Disciplines are used throughout the project, but are emphasized differently depending on the current lifecycle phase. Table 3-5 summarizes the activities of the RUP.

Table 3-5: RUP Summary

Lifecycle Phase	Description	Engineering Discipline	Description	Supporting Discipline	Description
Inception	Understand business case for product. Establish requirements, scope system.	Business modeling	Model organizational context for software.	Environment	Provide software development process and environment.
Elaboration	Use case modeling, architectural design, prototypes.	Requirements	Elicit stakeholder requirements.	Configuration and Change Management	Manage software changes.
Construction	Coding, unit and integration testing.	Analysis and Design	Show how system will be realized.	Project Management	Risk management, planning, and monitoring.
Transition	Beta-testing, production, training, maintenance.	Implementation	Implement and integrate components.		
		Test	Verify software's reliability, functionality, application performance, and system performance.		
		Deployment	Deliver and support software.		

3.1.6 Mapping Software Processes to the SE Framework

Table 3-6 shows the mapping of the generic and linear software processes to the categories and activities of the SE Framework.

Table 3-6: Mapping Generic and Linear Software Processes

SE Framework		Generic	Linear	
Category	**Activity**		**Waterfall**	**V-Model**
Requirements Development	Develop Stakeholder Requirements	Communication	Requirements Specification	Requirements Analysis
	Develop and Analyze System Requirements	Communication	Requirements Specification	Requirements Analysis
	Validate Requirements	Communication	Requirements Specification	Requirements Analysis
Conceptual Analysis	Develop Conceptual Model	Modeling	Analysis	--
	Validate Conceptual Model	Modeling	Analysis	--
Product Design	Perform Functional Analysis	Modeling	Design	System Design, Architecture Design
	Synthesize Design	Modeling	Design	Architecture Design, Module Design
	Verify Design	Modeling	Design	(All Design)
Product Development	Establish S/W Development Environment	Construction	Coding	Coding
	Implement Product Design	Construction	Coding	Coding
Product Testing	Perform Product Verification	Construction	Testing	Unit Testing, Integration Testing
	Perform Product Validation	Construction	Testing	System Testing, Acceptance Testing
Project Management Practices	Project Planning	Planning	--	Project Management Sub-model
	Project Control	Umbrella Activities	--	Project Management Sub-model
	Risk Management	Umbrella Activities	--	Project Management Sub-model
	Quality Management	Umbrella Activities	--	Quality Assurance Sub-model
	Configuration Management	Umbrella Activities	--	Configuration Management Sub-model
(Not Mapped)		Deployment	Operations System Requirements	

Table 3-7 shows the mapping of the incremental/iterative software processes to the categories and activities of the SE Framework. Software practitioners involved in the development of M&S tools should refer to the mapped activities from their chosen software engineering practices during the indicated activity in the SE Framework to find software engineering best practices applicable during that SE Framework activity.

Table 3-7: Mapping Incremental/Iterative Software Processes

SE Framework		Incremental/Iterative			
		Spiral		RUP	
Category	Activity	Iteration	Quadrant / Section	Phase	Discipline
Requirements Development	Develop Stakeholder Requirements	Spiral 1		Inception	Requirements
	Develop and Analyze System Requirements	Spiral 2		Inception	Requirements
	Validate Requirements	Spiral 2	IV	Inception	Requirements
Conceptual Analysis	Develop Conceptual Model	All Spirals	Prototyping	Elaboration	Business Modeling, Analysis and Design
	Validate Conceptual Model	--	--	Elaboration	Business Modeling, Analysis and Design
Product Design	Perform Functional Analysis	Spiral 3		Elaboration	Analysis and Design
	Synthesize Design	Spiral 3		Elaboration	Analysis and Design
	Verify Design	Spiral 3	IV	Elaboration	Analysis and Design
Product Development	Establish S/W Development Environment	Spiral 4		Construction	Environment
	Implement Product Design	Spiral 4	IV	Construction	Implementation
Product Testing	Perform Product Verification	Spiral 4	IV	Construction	Test
	Perform Product Validation	Spiral 4	IV	Transition	Test
Project Management Practices	Project Planning	All Spirals	III	All	Project Management: Planning
	Project Control	--	--	All	Project Management: Monitoring
	Risk Management	All Spirals	I	All	Project Management: Risk Management
	Quality Management	--	--	All	Test
	Configuration Management	--	--	All	Configuration and Change Management
(Not Mapped)		Spiral 0			Deployment

This page intentionally left blank.

4. CONCLUSIONS

The most notable observation about this effort is that, although there have now been decades of focus on engineering processes and process improvement, much of it has been focused on systems and software in general, not on models and simulations specifically, and much of it at the macro level, rarely daring to drill down to the level of individual best practices. The study team was surprised by the lack of detailed best practices for the development of models and simulations in the literature. The responses to the online questionnaire made it clear that the community generally viewed the request as unusual, although there were some thankfully notable exceptions. It is clear that there is a need for continued pursuit of this level of detail, both because of the overwhelmingly positive response to the work when it was presented to the community, and because there is good work being done in the area. However, few of the best practices have been well documented, as evidenced by the activities for which the team was unable to identify best practices. It is hoped that DoD will pursue development of a standardized guidance product based on this study within SISO, thereby leveraging the expertise of the community and engaging them in the successful adoption of the final product.

This page intentionally left blank.

APPENDIX A: REFERENCES

(a) ISO/IEC 15288-2008, "Systems and software engineering -- System life cycle processes," 2008.

(b) IEEE 1516.3-2003, "Federation Development and Execution Process," 2003.

(c) ANSI EIA-632, "Processes for Engineering a System," January 1999.

(d) IEEE 1220 [ISO/IEC 26702:2007(E)], "Standard for Application and Management of the Systems Engineering Process," 2007.

(e) MIL-STD-499C, "Systems Engineering," April 2005.

(f) *Systems Engineering Handbook* Version 3.1. *INCOSE*, August 2007

(g) CMMI-DEV. "CMMI for Development, Version 1.2." Pittsburgh, PA: Carnegie Mellon University Software Engineering Institute, August 2006.

(h) Acquisition Modeling and Simulation Master Plan (AMSMP), Systems Engineering Forum, Office of the Under Secretary of Defense (Acquisition, Technology and Logistics), 17 April 2006.

(i) CJCSI 3170.01, "Joint Capabilities Integration and Development System," 1 March 2009.

(j) DoD Directive 5000.1, "The Defense Acquisition System," 12 May 2003.

(k) W3C Working Group. "Relationship between Mobile Web Practices (MWBP) and Web Content Accessibility Guidelines (WCAG), 9 July 2009. Retrieved from http://www.w3.org/TR/mwbp-wcag/

(l) Pressman, Roger. Software Engineering: A Practitioner's Approach, 7th ed. New York, NY: McGraw-Hill, 2009.

(m) V-model (software development)." Wikipedia. 16 May 2010.

(n) Boehm, B. W. "A Spiral Model of Software Development and Enhancement." *Computer, 21*(5), 61-72, 1998.

(o) Boehm, B. W. "A spiral model of software development," *ACM SIGSOFT Software Engineering Notes, 11*(4), 14-24, August 1986.

(p) "IBM Rational Unified Process." Wikipedia. 18 May 2010.

This page intentionally left blank.

APPENDIX B: BEST PRACTICES DEFINITIONS

This appendix provides the complete descriptions of the best practices listed in Section 3 under each phase and activity. Practices are ordered as they are in Section 3. The citations referenced in the Author(s) column can be found in Appendix D. In some cases, a practice resulted from the merger of similar practices from more than one source. In these cases, there are multiple references in the Author(s) column.

Title	Definition	Author(s)
Establish intent for model use	When preparing to build a model, developers should establish the intended use for the model by asking the following questions: • What is the system's role within the system, such as an "oracle" advisor, or part of some converging evidence in a voting scheme? • What is the model's place in the system, such as part of the user interface, visual display, statistical analysis, or animator? • What is the model's viewpoint, such as the user's viewpoint in a training environment or a device's point of view in a large system model that includes many models? • What do I know about the problem and the possible solutions? • What are the important aspects of the phenomenon? • What is it supposed to do? • How do we handle different populations of users and types of data? • As we scope the modeling activity, what can we simplify and what level of detail and precision is required? • When we envision the use of a model, how will it interact with its environment? • How will its users (if any) interact with it during processing? • How will its users determine (or measure) what it does and whether it is successful?	Kirstie Bellman and Christopher Landauer [3]
Use focus groups in simulation creation	Simulation technology projects can be effectively organized around focus groups, especially in high-payoff areas. The group should emphasize cross-domain collaboration and should be closely tied with the operational communities the simulations will support.	W. H. (Dell) Lunceford Jr. [34]

Title	Definition	Author(s)
Specify data content	A formal requirement for environmental data as either the input or output of an M&S system or a program that supports them is necessary for simulations that are dependent on environmental data. A data model and associated toolset that allows for this concept promotes the specification of requirements within the data and allows tools to validate environmental data. An example of such a framework is Synthetic Environment Data Representation and Interchange Specification (SEDRIS) and its associated toolset.	Jesse Campos, Greg Hull, and Farid Mamaghani [8]
Use survey methods to elicit SME knowledge	Techniques for building objective simulation referents from the knowledge of SMEs can be improved in the following manner. In an experiment a conventional survey regimen is used for constructing a questionnaire to sample SME opinion on the conditions and outcomes of an envisioned scenario. This initial questionnaire is used for vetting and is given to a small set of SMEs in structured interviews. These results of these interviews may improve the quality of the questionnaire given to a larger set of SMEs but also decreased the number of questions.	Scott Harmon, Mike Metz, and Simone Youngblood [22]
Establish model focus by carefully choosing model behavioral aspects and data	When designing a model, the architect should ask the following questions: • What to model, or how to choose the important aspects of a system? • How to select the appropriate partial models of behavior of the system and its environment? • What are the problem's important phenomena? • How flexible and variable must the model be? • What are the sources of relevant data and their models? A good model must: • Account for the behavior that is important to the problem • Provide ways of learning what it does and how it works • Need no more detailed information to run or to explain than the level of detail for the problem	Kirstie Bellman and Christopher Landauer [3]
Use a formal language for linking requirements to a conceptual model	UML/SysML diagrams may be used for requirements analysis and conceptual modeling. Requirements and use case diagrams can be used to explore objectives. Activity, sequence, state machine, block definition, and parametric diagrams can model dynamic behavior traceable back to the requirements.	Michel Keuning and Arno Gerretsen [31]

Title	Definition	Author(s)
Use a standard process for creating a conceptual model	A Conceptual Model (CM) should be constructed in the following manner. 1) Define use cases 2) Selectively define process and data flow models 3) Define major concepts as classes 4) Define static relationships between classes (Subtypes, aggregations, composition, and general associations) 5) Define key attributes for the classes 6) Define methods that link the classes 7) Define dynamic relationships between classes 8) Define representative scenarios 9) Do Gap analysis: complete class diagrams with attributes and methods 10) Develop additional classes if necessary Capture concepts by brainstorming with SMEs and documenting in formal languages such as UML. The CM can become "the basis for a standardized ontology framework."	Judith L. Cerenzia, et al. [9]
Select computer scientists with domain expertise to be on the conceptual modeling team	Good conceptual modelers are computer scientists with domain expertise. If either skill is lacking, the conceptual modeler has difficulty building a model that bridges the gap between the real world and the computation space.	Clark R. Karr [29]
Augment logical data models with semantics	Using a logical data model to describe a canonical form for input is a good idea. Commonality in data is the foundation for interoperability at a syntactic level (i.e., data can be exchanged in standard formats). For higher levels of interoperability, not only the data but also its context needs to be standardized through a common reference model, followed by commonality of usage (algorithms and logical inference) for knowledge-level interoperability. An ontology can provide the vocabulary and necessary conceptual interrelationships to permit greater automation in data interchange and data processing.	Curtis Blais, et al [5]
Create a data dictionary	Create a data dictionary to define elements, specify primitive behaviors to be represented in the model, and ensure consistent use of terms. Data dictionaries include terms used for classifications, attributes, operations, values, metadata, units, etc. The data dictionary should be reusable for different M&S projects; each M&S project may extend the data with specific terms as needed. Highly specific extensions may not be integrated into the common data dictionary but may reside with the logical model for that particular M&S application. A semantic ontology can be used to define primitive and composite behaviors to be represented in models.	Dale Miller, et al. [37]; William J. Gerber and Lee W. Lacy [20]

Title	Definition	Author(s)
Include full simulation specification and context in a conceptual model for a simulation system	Include the following information in a simulation conceptual model: 1) Simulation descriptive information: model identification (e.g., version and date); Points-of-Contact (POC)s; model change history 2) Simulation context (per intended application): purpose and intended use statements; pointer to M&S requirements documentation; overview of intended application; pointer to sources of application domain information; constraints, limitations, assumptions; pointer to referent(s) and referent information 3) Simulation concept (per intended application): mission space representation (simulation elements and simulation development description); simulation space functionality 4) Simulation elements, including: entity definitions (entity description, states, behaviors, interactions, events, factors, assumptions, constraints, etc.); process definitions (process description, parameters, algorithms, data needs, assumptions, constraints, etc.); natural environment representations 5) Validation history, including: M&S requirements and objectives addressed in Verification and Verification (V&V) effort(s); pointer to validation report(s); pointer to simulation conceptual model assessment(s) 6) Summary: existing conceptual model limitations (for intended application); list of existing conceptual model capabilities; conceptual model development plan.	Dale Pace [42]; Virginia Dobey and Paul Foley [15]

Title	Definition	Author(s)
Format the conceptual model using a standard notation accessible to all stakeholders	The format of conceptual model documentation should accomplish two objectives: 1) Ensure that the simulation design team fully understands what the simulation must do so that an appropriate simulation design can be developed, and 2) Facilitate communication among all simulation stakeholders so that all fully understand simulation capabilities, limitations, and assumptions. It should be remembered that the stakeholders include the simulation development team and simulation users, those involved in assessing the simulation (such as V&V personnel), SMEs used in simulation development and/or assessment, those impacted by results from the simulation, simulation sponsors, and perhaps others. One approach to documenting simulation-related conceptual models is the design accommodation method. With this approach, the simulation developer uses a descriptive format, such as Unified Modeling Language (UML), that has been chosen to support simulation design to describe and document the conceptual model. There are advantages to such an approach: • It minimizes the opportunity for misunderstanding and error as the simulation developer transforms the conceptual model into the simulation design. • It facilitates keeping the conceptual model current with evolution of the simulation.	Dale Pace [42]
Combine conceptual modeling with knowledge acquisition/knowledge engineering	Combine conceptual modeling with knowledge acquisition/knowledge engineering (KA/KE). Conceptual modeling is an iterative and repetitive process that determines what knowledge need be acquired and then engineered, and should be revised throughout a modeling study. The model design includes both the conceptual model and the design of the code.	Clark R. Karr [29]; Stewart Robinson [48]
Document a rationale for realistic output measures	There must be a firm rationale for measures that are calculated from input data and the model. Begin by questioning what scientific theory, functional relationships, and data exist to calculate a particular output measure.	Bart Bennett, Richard Hillestad, and Gordon Long [4]
Use a standardized conceptual model to mitigate stakeholder subjectivity in simulation design	The quality of a simulation is strongly determined by its designers yet individual designers may have different thought processes for any given simulation. Such subjectivity in modeling can be mitigated by guidelines to support the traceability to a conceptual model. Such conceptualizations of the simulation and support for their representations in an information model are essential. Often using a standardized design tool such as UML will enable this process. These guidelines for modeling in addition to standardization efforts take into account the interrelation between the problem to be modeled and its representation.	Reinhard Schuette and Thomas Rotthowe [52]

Title	Definition	Author(s)
Involve the decision-maker in the model development process	The decision maker is the primary user of the model. The decision maker must be involved in the model development process, particularly the conceptual modeling phase. High output resolution and sophisticated graphic effects do not imply high validity, but they do tend to lend perceived credibility to the simulation. The decision maker or end user needs to understand how the simulation is built to have "an appropriate degree of confidence in the simulation and ultimately its outputs."	Paul A. Roman [50]
Maintain a distinction between models and simulations	An effective, composable model must be clearly distinguishable from its environment as well as from controller components. Models should strive to represent the functionality and the phenomenology of the systems being represented rather than any specific simulation environment in which they may be ultimately utilized or controlled, i.e., models should be distinct from and have a clean interface with the simulation kernel.	James C. Watkins, et al. [63]
Use archived models and model components from an authoritative source	Reuse other models and components where possible in the construction of simulations. However, users of previously-developed models must be careful to understand the effects and possible ramifications of using such models. Defined interfaces and adequate documentation are necessary.	James C. Watkins, et al. [63]
Select input data items based on a complete problem context	When designing data production and retrieval systems, consider: • Contextual information - the "who," "where," and "when" of the scenario • Objectives - "why" of the scenarios • Strategies, plans, and tactics - "how" of the scenario, including behaviors • Quantity and type of resources - "what" of the scenario • Characteristics and performance - an extension of the "what" used to describe a system's abilities • Environment - interaction between systems and nature • Costs - resource expenditure including money, time, and people	Bart Bennett, Richard Hillestad, and Gordon Long [4]
Define uncertainty models	Define the uncertainty models that describe the limits of knowledge about the model. Uncertainty models range from so-called confidence factors to belief functions, empirical probabilities, chaos, colored noise, and fuzzy sets.	Kirstie Bellman and Christopher Landauer [3]
Use design patterns in M&S	Patterns and pattern languages can be used to describe successful solutions to common software problems. They should be used where possible in the development of M&S applications.	Kenn Atkinson [1]

Title	Definition	Author(s)
Balance modeling needs with data considerations	When selecting a model's structure, particularly whether certain aspects will be modeled via code or input data, consider the nature of the data and the challenges associated with creating and managing databases. Selection of a convenient algorithm should not be done without considering its data needs.	Bart Bennett, Richard Hillestad, and Gordon Long [4]
Design data storage and retrieval architecture	Data architecture hardware and software needs to provide fast, reliable access to data: • For complex multi-tasking and multi-programming activities, • To be able to update databases rapidly and accurately, with little dedicated effort, and • To help verify, validate, and accredit a model.	Bart Bennett, Richard Hillestad, and Gordon Long [4]
Consider availability of data sources when designing simulation	Data sources, availability, and accessibility constrain simulation design. The following sources should be considered: <u>Histories</u>: • Useful for typifying conditions, systems, processes • Show what has occurred-Include human decisions • May not capture enough detail • May contain plausible events that never occurred • May not apply to the future <u>Training or Exercises</u>: • Instrumented ranges provide a lot of quantitative data • Can examine specific conditions • Include human decisions but may contain biases • Emphasize training objectives and safety, so may introduce biases • Can't capture some events • Human learning prohibits consistently repeatable results <u>Tests:</u> • Rigorously measures parts of the system • Control and classify experimental conditions • Capture characteristics and performance data • Have a limited scope • Use narrowly-defined conditions • Emphasize test objectives and safety, so may introduce biases	Bart Bennett, Richard Hillestad, and Gordon Long [4]

Title	Definition	Author(s)
Consider availability of data sources when designing simulation (cont'd)	Experts: • Often necessary for extrapolating to the future • Good sources for strategies and tactics • May be the only sources for some information • Not rigorously measureable or standardized • Tend toward anecdotal information • May have organizational, institutional, or other biases Other models: • Help reduce the overall context, breadth, or detail • Capture data from other sources above • May help automate data processing • Often difficult to join multiple models conceptually and physically • Difficult to aggregate and decompose	Bart Bennett, Richard Hillestad, and Gordon Long [4]
Group and separate data from models with varying resolution	If the simulation depends on data from models of higher or lower resolution, it must appropriately process this data by adjusting for formats and metrics and by properly grouping (aggregating) and separating (disaggregating) the information. Aggregation and disaggregation affect certain simulation inputs including time, geography, objects, and measures which are partially interdependent.	Bart Bennett, Richard Hillestad, and Gordon Long [4]
Distinguish unknowns from unknowables	Simulation designers must distinguish and document information that is unknowable from that which is merely unknown in the data and then use varying precisions, algorithmic processing, and functional forms to match the data's uncertainty and availability. Efforts to analyze the unknowns may produce concrete solutions that eliminate future concerns, but efforts aimed at that which is unknowable will yield only subjective interpretations. Still, both may be necessary to make important data available to a simulation.	Bart Bennett, Richard Hillestad, and Gordon Long[4]

Title	Definition	Author(s)
Use intelligent analytical approaches to handle unavailable or unknowable data	When data is unavailable or unknowable, try applying these analytical approaches to help a simulation use the data despite its deficiencies: • Best estimates: Select various best estimates for values that an unavailable data item can take on. • Boundary analysis: Instead of using one or a few best guesses, examine the likely maximum and minimum values to determine what the unavailable data item will do. • One-sided arguments: To greatly reduce the data values to be examined, create an argument with values that clearly exceed or underestimate the potential actual values and observe whether the results meet requirements. • Analyzing parameters or sensitivities and exploratory modeling: Take discrete values across a data item's potential range to determine how one value affects another. • Distributional analysis: Use a probability distribution to define the data and Monte Carlo methods to select the actual data values for particular cases.	Bart Bennett, Richard Hillestad, and Gordon Long [4]
Adopt commonly accepted icons, symbols, shapes, and colors used to represent simulation entities, where possible	When designing simulation Graphical User Interfaces (GUIs), use icons, symbols, shapes, and colors that are familiar representations of simulation entities.	Jim Wall, Randy Elms, and Dave Nock [61]
Evaluate a model's pedigree before (re)using it as a component	In composing a simulation one should know the following about the models that are being re-used (composed): 1) Mathematical origins 2) Range of applicability 3) Environment in which the model is intended to be run 4) Connections (i.e., inputs and outputs and their formats) Modeling tools such as UML can and should be used to capture the pedigree of a model.	Brian Goldiez [21]

Title	Definition	Author(s)
Create both an analysis data model and a logging data model to facilitate capture and use of simulation output data	Create an analysis data model that defines captured data in a suitable form for analysts and decision makers. At a lower level of abstraction, create a logging data model that defines the data that will be logged during the simulation run. It is also useful to define a mapping or translation process to aggregate the logged data to facilitate analysis.	Ke-Thia Yao [65]
Use standards where applicable	Application-level standards [Open Systems Interconnections (OSI) Model Level 7] like XML, XMI, Extensible Stylesheet Language Transformation (XSLT), UML, and ISO Standard for the Exchange of Product (STEP) model data should be used whenever possible because their flexibility allows them to be used in unison in M&S applications.	James W. Hollenbach [26]
Separate data Input/Output (I/O) interface from model code	Establish well-defined interfaces between the model code and the input/output data (initial conditions, factors, and results) for the model.	Stewart Robinson [49]; James C. Watkins, et al. [63]
Use a standardized logical data model and format for I/O data	Use a logical data model in a standard format (such as XML) to describe a canonical form for input.	Michael Scott Jacobs [27]; John Schloman [51]
Select output data items based on a complete view of simulation usage	The following types of output data items should be considered: • Measures - the simulation's main output used to determine if it meets objectives • Intermediate measures - more detailed outputs that help to determine if the measures are being calculated correctly • Histories - time-stamped chronicle of events • Repeated inputs - data that ties results to the input that created it • Diagnostics - information needed to determine and test how a simulation should run • Graphics - visual displays With the goal of facilitating: • Determination of cause-and-effect relationships • Identification of individual simulation runs • Flow across replications and cases • Flow to the input of another model	Bart Bennett, Richard Hillestad, and Gordon Long [4]

Title	Definition	Author(s)
Design models as components with loose coupling	Models should be viewed and developed as collections of components that are designed and implemented for reuse.	James C. Watkins, et al. [63]; Osman Balci [2]; Phillip E. Pournelle, Curtis Blais, and Don Brutzman [45]
Use scenario generation tools to promote consistency and efficiency	Scenario generation tools promote consistency and efficiency. Using scenario generation tools to automate the specification of order of battle, mission tasking, model parameters, environmental descriptions and so on promotes consistency across runs and leads to timely simulation runs and analysis.	Mark Kilby and Laurence Esmonde [32]
Choose the right architecting tool for static and dynamic aspects of the M&S application	Architecture definition tools (e.g., modeling frameworks and environments) can potentially help address DoD's architecting challenge associated with M&S applications. The use of DoDAF Architecture views for static aspects of the problem being modeled as well as Object Modeling Group's (OMG's) data-driven architecture-based Model Driven Architecture (MDA) are examples of two such tools for M&S applications.	James W. Hollenbach [26]
Employ common random numbers in models	Reference pseudo-random number streams only once per model. Using common random numbers ensures that random sampling is replicated exactly between cases and that variability between cases is attributable to changed factors, not to different random number draws.	Stewart Robinson [49]
Collect referent information	Identify and collect appropriate referent information for validation of M&S results (test and experimental data and observations, laws of physics and theory, results from other M&S, SMEs, etc.) with explicit quantification of uncertainties related to that information.	Dale Pace [43]
Decompose qualitative SME input into quantitative indicators	Qualitative concepts used by SMEs may be decomposed hierarchically in a recursive fashion until all sub-components (called indicators) are quantitative. These quantitative indicators (called leafs) can be combined with weights to measure conformance. This allows comparisons of qualitative concepts.	Osman Balci [2]
Validate models against each intended use	Models must be validated against each intended use. If a previously-validated model is applied to new questions or new uses, it must be reevaluated in the new context.	Paul A. Roman [50]
Conduct engineering integration reviews	Engineering Integration Reviews (EIRs) are used to better understand the impacts that specific code integration may have on system operation.	Doug Parsons [44]

Title	Definition	Author(s)
Include user domain representatives and external developers in peer reviews	Peer reviews are conducted throughout the software development process. Representatives of user domains and external developers are invited to review and write defects, as appropriate, from artifacts in these phases. These phases include requirements analysis, software design, and software integration and test.	Doug Parsons [44]
Use SMEs throughout the development life cycle	SMEs should be employed to cover all areas of the problem domain and all phases of the development life cycle. A different SME may be selected for a different phase of development life cycle such as requirements, design, implementation and integration.	Osman Balci [2]
Use Systems Engineering analysis and documentation	As part of any systems engineering-based M&S, there should be documentation of both the static and dynamic M&S aspects to improve model comprehensibility. These should be associated with model ventilation; a term for the exercise in the analysis of a model's assumptions, deficiencies and limitations.	Tuncer Oren [40]
Use a standardized method of "packaging" for developing model components	Use a standardized file structure to facilitate extraction and reintegration of model components.	James C. Watkins, et al. [63]
Document model abstraction decisions	When designing the model abstraction, document choices, reasoning and assumptions. Write down the assumptions under which the selected model simplifications are valid. State clearly the reasons for choosing certain representations and methods; this includes writing down limits and expectations for the representations. Corresponding to each assumption are criteria to determine if it is no longer valid. Check these criteria occasionally and every time you use the simulation.	Kirstie Bellman and Christopher Landauer [3]
Keep data current	When collecting data, it should be tagged with its source and last update date/time. This allows currency to be determined and easy checks for updates.	Bart Bennett, Richard Hillestad, and Gordon Long [4]
Establish a configuration management system	Establish an effective configuration management system so that each variation of the code is distinctly identified and each use of the code is associated with a particular code variation; the conceptual model related to a variation should be identified in a way that associates them.	Dale Pace [43]

Title	Definition	Author(s)
Document models and simulation data with metadata	Metadata about model and simulation components as well as simulation runs is critical to the proper use and interpretation of data as well as the efficient use and reuse of models. The UML and the Extensible Mark-up Language (XML) Metadata Interchange (XMI) are examples of languages that can be used to represent metadata. Both syntactic and semantic attributes can be described using UML and its associated extensions promoting reuse of model components at a conceptual level. Metadata about input to and output from a simulation run might include the metrics associated with data items, formats, sources, assumptions, restrictions on distribution, its currency, reliability, usage limits, the relationship to other data, history of changes, application, and other useful comments.	Stewart Robinson [49]; Bart Bennett, Richard Hillestad, and Gordon Long [4]

This page intentionally left blank.

APPENDIX C: SIDE-BY-SIDE SE FRAMEWORK COMPARISON

Table C-1: Requirements Development

ANSI/EIA-632	There are three processes associated with the requirements definition phase that determine (1) acquirer requirements; (2) other stakeholder requirements; and (3) system technical requirements. The first two steps include identifying, collecting, and defining acquirer and other stakeholder requirements. The third step in the requirements determination phase transforms the validated set of requirements into a set of validated system technical requirements and assumptions. The outcome of this phase coupled with the solution definition phase (product design) produce the system design - specifications, drawings, models.
ISO/IEC-15288	In this standard, the requirements development phase of the technical process maps to the stakeholder requirements definition process. The purpose of the stakeholder requirements definition process is to define the requirements for a system that can provide the services needed by users and other stakeholders in a defined environment. It identifies stakeholders involved with the system throughout its life cycle, and their needs, expectations, and desires. It analyzes and transforms these into a common set of stakeholder requirements that express intended interaction the system will have with its operational environment and that are the reference against which each resulting operational service is validated. In addition to the development of the stakeholder requirements, system requirements need to be defined. This step includes such activities as the definition of system characteristics, attributes, and functional and performance requirements. In addition, constraints that will affect the architectural design of a system and the means to realize it are specified. The integrity and traceability of system requirements to stakeholder requirements must be defined. And finally, a basis for verifying that the system requirements are satisfied is defined.
ISO/IEC-26702	In this particular standard, the requirements analysis and validation stages describe the overall requirements development process within the MSDBP SE framework. The requirements analysis stage establishes what the system will be capable of accomplishing; how well system products are to perform in quantitative and measurable terms; the environments in which system products operate; the requirements of the human/system interfaces; the physical characteristics; and constraints that affect design solutions. This forms the requirements baseline and analyses are conducted to resolve any conflicts. The requirements validation stage evaluates the requirements baseline to ensure it represents stakeholders' expectations and constraints. In addition, this stage assesses the requirements baseline to determine whether the full spectrum of possible system operations and system life cycle support concepts has been adequately addressed.
IEEE 1516.3 (FEDEP)	The purpose of this phase of the process is to define and document a set of needs that are to be addressed through the development and execution of the project and to transform these needs into a more detailed list of specific objectives. In defining user/sponsor needs, a clear understanding of the problem needs to be developed. Next is to document what must be accomplished to achieve those objectives.

Table C-1: Requirements Development (continued)

MIL-STD-499C	In this standard, requirements analysis and validation is conducted to develop the system technical requirements and constraints. The requirements baseline is the product of the requirements development phase. In this phase of the process, the requirements baseline shall trace to the capabilities for which the system is being designed and to the missions for which it is intended. It shall encompass the minimum operator/user capabilities to be and balance the capabilities with cost, schedule, risk, and potential for evolutionary growth. In addition, the requirements baseline shall include system interoperability. It shall include all functional and performance requirements and constraints and those imposed by each specialty function. The requirements baseline shall include all constraints and design constraints for interoperability, security, safety, human factors, reliability, maintainability, and other relevant constraint categories. The requirements baseline shall be documented.
INCOSE Handbook (v3)	The purpose of the stakeholder requirements definition process is to elicit, negotiate, document, and maintain stakeholders' requirements for the system-of-interest within a defined environment. The process includes several activities such as to identify stakeholders, elicit requirements, and define constraints. Other activities include establishing critical and desired system performance, establishing measures of effectiveness and analyzing requirements for clarity, completeness, and consistency. Also, the activities include negotiating modifications to resolve unrealizable requirements; validate, record, and maintain requirements throughout the entire process; and establish and maintain a traceability matrix.
CMMI-DEV	This process area describes customer requirements and product requirements. Taken together, these requirements address the needs of relevant stakeholders and product attributes. Requirements also address constraints caused by the selection of design solutions. Stakeholder needs, expectations, constraints, and interfaces are collected and translated into customer requirements. Customer requirements are refined and elaborated to develop product and product component requirements.
MSDBP	The purpose of this phase of the M&S development process is to produce the set of requirements that will drive M&S design activities and provide the criteria by which the success of the M&S development project will be judged. This includes all categories of requirements, and all activities needed to ensure completeness and consistency of the requirements throughout the product lifecycle. Although requirements may be refined during any stage of the development process, all M&S best practices related to any aspect of requirements development, analysis, or validation will be captured in this section.

Table C-2: Conceptual Analysis

ANSI/EIA-632	This phase of the overall technical process is part of the Product Design (Solution Definition Process), but not a separate phase in the process.
ISO/IEC-15288	In this particular standard, the requirements analysis process closely corresponds to the conceptual analysis technical process for the MSDBP SE framework. The purpose of the requirements analysis process is to transform the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services. This process builds a representation of a future system that will meet stakeholder requirements and that, as far as constraints permit, does not imply any specific implementation. It results in measureable system requirements that specify, from the supplier's perspective, what characteristics it is to possess and with what magnitude in order to satisfy stakeholder requirements.
ISO/IEC-26702	In this particular standard, Clause 6 describes the SE processes and Clause 5 describes the application of the processes. The functional analysis and verification stages of Clause 6 best describe the conceptual analysis process within the MSDBP SE framework. The system definition stage of Clause 5 best describes the conceptual analysis process. Functional analysis is accomplished by translating the validated requirements baseline into a functional architecture. The functional architecture describes the functional arrangements and sequencing of subfunctions resulting from decomposing the set of system functions to their subfunctions. Functional analysis should be performed without consideration for a design solution. Functional verification is conducted to assess the completeness of the functional architecture in satisfying the validated requirements baseline and to produce a verified functional architecture for input to synthesis. The system definition stage establishes the definition of the system with a focus on system products required to satisfy operational requirements. The major events of this stage should include completion of system, product, and subsystem interface specifications, system and product specifications, and preliminary subsystem specifications; establishment of a system baseline; and completion of technical reviews appropriate to this stage.
IEEE 1516.3 (FEDEP)	The purpose of this step of the FEDEP is to develop an appropriate representation of the real world domain. In this step, objectives are transformed into a set of highly-specific requirements that will be used in the design, development, testing, execution, and evaluation. There are three main steps in the conceptual analysis process: (1) develop a functional specification of the scenario; (2) produce a conceptual representation of the intended problem space base on the interpretation of user needs and objectives which is an implementation-independent representation; and (3) develop and define detailed requirements which are testable and provide the implementation level guidance needed to design and develop the project.

Table C-2: Conceptual Analysis (continued)

MIL-STD-499C	Functional analysis is conducted to develop a functional architecture or logical representation of the system. The functional architecture shall accurately and completely reflect the functional and performance requirements in the requirements baseline. It shall accurately and completely reflect the minimum or threshold required operational capabilities before commencement of detailed design. In addition, the functional architecture at each level shall be sufficiently defined to form the basis for detailed and precise functions or logical elements and their allocated performance/ functional requirements at the next lower level. Each top-level function shall be decomposed to lower levels with sufficient detail. Each decision in the functional architecture that is chosen shall be supported by documented trade-off or other analysis. The documentation shall be maintained in a decision database. Data flow relationship shall be established. Interfaces shall be defined at the earliest possible time and to as great a detail as possible.
INCOSE Handbook (v3)	For this standard, the conceptual analysis process is comprised of the output of the requirements definition phase and the activities in the requirements analysis phase. The output of the requirements definition phase consists of formally documented and approved stakeholder requirements that will govern the project, including: required system capabilities, functions and/or services; quality standards; cost and schedule constraints; concept of operations; and concept of support. The outputs should include measures of effectiveness and suitability that will be used for assessing the realized system and enabling systems. Validation criteria may specify who will perform validation activities, and the environments of the system-of-interest and enabling systems. Other outputs establish the initial baseline for project scope and associated agreements. The purpose of the Requirements Analysis Process is to review, assess, prioritize, and balance all stakeholder and derived requirements (including constraints); and to transform those requirements into a functional and technical view of a system description capable of meeting the stakeholders' needs. This view can be expressed in a specification, set of drawings or any other means that provides effective communication. The output of this phase of the process is a technical description of characteristics of the future system must have in order to meet stakeholder requirements - not a specific solution - which will be evolved in subsequent development processes. It derives additional requirements resulting from analysis of the input stakeholder requirements as required to meet project and design constraints; defines the functional boundaries for the system to be developed; and identifies and documents any interfaces and information exchange requirements with systems external to the functional boundaries. The total set of requirements encompasses the functional, performance, and non-functional requirements and the architectural constraints. Any decisions taken are documented in the information repository.
CMMI-DEV	Product component requirement development taken with developed customer and product requirement form part of the conceptual analysis process in the MSDBP SE process. Customer requirements are analyzed in conjunction with the development of the operational concept to derive more detailed and precise sets of requirements called "product and product component requirements." The requirements are allocated to product functions and product components including objects, people, and processes. The traceability of requirements to functions, objects, tests, issues, or other entities is documented. The allocated requirements and functions are the basis for the synthesis of the technical solution. As internal components are developed, additional interfaces are defined and interface requirements are established. The requirements are analyzed and validated, and a definition of required functionality is developed. The other process area discussed in this particular standard that makes up conceptual analysis is the management of requirements. The purpose of Requirements Management is to manage the requirements of the project's products and product components and to identify inconsistencies between those requirements and the project's plans and work products.

Table C-2: Conceptual Analysis (continued)

MSDBP	The purpose of this phase of the M&S development process is to produce an implementation-independent conceptual depiction of the real world missions and operations that must be represented in the desired M&S application. The product resulting from this activity is generally referred to as a conceptual model. This model can be used as the structural basis for many design and development activities and can highlight correctable problems early in the development of the M&S application when properly validated by the appropriate stakeholders.

Table C-3: Product Design

ANSI/EIA-632	The Solution Definition Process is used to generate an acceptable design solution. This solution satisfies: 1) the system technical requirements resulting from completing the Requirements Definition Process and 2) the derived technical requirements from the Solution Definition Process. The three requirements associated with the Solution Definition Process are the development of logical solution representations, physical solution representations, and specified requirements. The first process is the development of a validated set of logical solution representations or more specifically, an abstract definition of the solution. This is also known as conceptual analysis. This process is not a separate phase of this particular systems engineering process, but is part of the solution definition or product design phase. Once this process is complete and is combined with the system technical requirements, the next step is to develop physical solution representations. The physical solution representations characterize the product/system design. This is an iterative process. The last step of the solution definition or product design is to specify requirements for the design solution. This includes specifying functional and performance requirements, physical characteristics, and test requirements.
ISO/IEC-15288	The purpose of the architectural design process is to synthesize a solution that satisfies system requirements. This process summarizes and defines areas of solution expressed as a set of separate problems of manageable, conceptual, and realizable proportions. It identifies and explores one or more implementation strategies at a level of detail consistent with the system's technical and commercial requirements and risks. An architectural design solution or a specified design requirement is defined in terms of the requirements for the set of system elements from which the system is configured. The product of this process is an implementable and traceable architectural design that satisfies validated requirements and is the basis for integration.

Table C-3: Product Design (continued)

ISO/IEC-26702	In this particular standard, Clause 6 describes the SE processes and Clause 5 describes the application of the processes. The synthesis and design verification stages of Clause 6 best describe the product design process within the MSDBP SE framework. The preliminary design and detailed design stages of Clause 5 best describe the product design. Synthesis tasks are performed for the purpose of defining design solutions and identifying subsystems to satisfy the requirements of the verified functional architecture. Synthesis translates the functional architecture into a design architecture that provides an arrangement of system elements, their composition, interfaces, and design constraints. The activities of synthesis involve selecting a preferred solution from a set of alternatives and understanding the associated cost, schedule, performance, and risk implications. Design verification is performed for the purpose of assuring that requirements are traceable to the verified functional architecture and the design architecture satisfies the validated requirements baseline. The preliminary design stage in Clause 5 is executed to initiate subsystem design and create subsystem-level specifications and design-to-baselines to guide component development. The final preliminary design stage documents should include identification of recommended components and interfaces; resolution of subsystem-level risks; assessment of component risks; and design for quality factors. The detailed design stage is executed to complete the subsystem design down to the lowest component level and create a component specification and build-to component baseline for each component.
IEEE 1516.3 (FEDEP)	The purpose of this step of the FEDEP is to produce the design of the project. This involves identifying components and creating participants, allocating functionality, and developing a detailed plan for development and implementation.
MIL-STD-499C	In this standard, the allocated baseline and design or physical solution representation comprise the product design phase of the MSDBP process. The allocated baseline shall include the physical hierarchy that identifies all system products, and shall establish the interactions of the system. It shall include the design-to technical functional and performance requirements and design constraints for each product in the physical hierarchy allocated such that requirements baselines will be fully satisfied over the system life cycle. The allocated baseline shall include all derived design-to requirements and design constraints for each product in the physical hierarchy. It shall include all interfaces that shall be defined at the earliest possible time and to as great a detail as is possible. In addition, in defining interfaces, how the interface will be physically implemented, as well as the logical issues such as data formats, data semantics, etc., shall be addressed. It shall include a verification method of analysis, inspection, demonstration, or test selected for each requirement and constraint. The design representation shall develop and assess alternative solutions; identify and quantify decision criteria; and analyze decision uncertainties. It shall perform the required functions within the limits of the performance parameters prescribed, identify constraints, and represent a balanced solution. It shall be designed for interoperability. Design representations shall be based on how well the solutions meet operational effectiveness measures along with constraints. Mature technologies and open architecture shall be considered. Opportunities for designing items for re-use shall be identified. Computer resources for system end items as an integral part of overall systems development shall be managed. The design representation shall include internal and external interfaces. It shall include products, processes, operational concepts, configurations, and people. The design representation shall evaluate alternatives, shall allow for tolerances and variations in the design while still meeting needed system capabilities and requirements, and it shall be traced to the allocated baseline.

Table C-3: Product Design (continued)

INCOSE Handbook (v3)	In this standard, architectural design maps to the product design in the MSDBP SE process and its purpose is to synthesize a system architecture baseline that satisfies the requirements. Architectural design begins from the baseline functional and performance requirements, architectural constraints, and traceability matrix. Specifications for enabling systems are used to drive interface design. Specifications for reusable system elements are used when designing for product lines. The result of this process is an architectural design that is placed under configuration management. This baseline includes system element detailed descriptions, requirements assigned to system elements and documented in a traceability matrix, and interface requirements and a plan for system integration and verification strategy.
CMMI-DEV	The Technical Solution process area focuses on evaluating and selecting solutions (sometimes referred to as "design approaches," "design concepts," or "preliminary designs") that potentially satisfy an appropriate set of allocated requirements and developing detailed designs for the selected solutions (detailed in the context of containing all the information needed to manufacture, code, or otherwise implement the design as a product or product component). This process area maps to the product design process of the MSDBP SE process.
MSDBP	The purpose of this phase of the M&S development process is to produce the design of the M&S application. This is normally conducted in an iterative fashion, with multiple loops of analysis, synthesis, and verification, resulting in the design of the system architecture for development. The number of design loops is primarily driven by the size and complexity of the M&S application, as dictated by the system requirements.

Table C-4: Product Development

ANSI/EIA-632	In general, this phase of the technical process transforms the product design or characterized design solution into a fully integrated end product conforming to specified requirements. In addition, this phase transitions verified products (M&S applications) to the acquirer for use. For this standard, the technical process of product development is referred to as the product realization process. The product realization processes are used to: (1) convert the specified requirements and other design solution characterizations into either a verified end product or a set of end products in accordance with the agreement and other stakeholder requirements; (2) deliver these to designated operating, customer, or storage sites; (3) install these at designated operating sites or into designated platforms; and (4) provide inservice support, as called for in an agreement. The two processes related to this phase of the development are implementation and transition to use processes. The implementation step transforms the design in accordance with the specified requirements to obtain a verified end product. The transition to use step results in products delivered to the appropriate destinations, in the required condition for use by the acquirer, and for the appropriate training of installers, operators, or maintainers of the products. This is in accordance with any established agreements/requirements.

Table C-4: Product Development (continued)

ISO/IEC-15288	In this standard, there are three processes that map to the Product Development process for the MSDBP. They are the Implementation, Integration, and Transition to Use processes. The purpose of the implementation process is to realize a specified system element. This process transforms specified behavior, interfaces and implementation constraints into fabrication actions. It results in a system element that satisfies specified design requirements through verification and stakeholder requirements through validation. The purpose of the integration process is to assemble a system that is consistent with the architectural design. The products of this process are a system integration strategy, definition of unavoidable constraints, and an integrated system capable of being verified against specified requirements. The purpose of the transition to use process is to establish a capability to provide services specified by stakeholder requirements in the operational environment. The process installs a verified system, together with relevant enabling systems, as defined in agreements. This process is used at each level in the system structure and in each stage to complete the criteria established for exiting the stage.
ISO/IEC-26702	In this particular standard, fabrication, assembly, and integration stages make up the product development process. The activities for this stage include fabricating hardware and implementing software components. At each step of this stage, testing is conducted to determine if the system or product fails to satisfy requirements. In addition, applicable technical reviews should be conducted to assess the maturity of the development effort.
IEEE 1516.3 (FEDEP)	The purpose of this step of the FEDEP is to develop the product, modify participants and components as necessary, and prepare the product for integration and testing. Key activities include developing the product, establishing agreements, implementing designs, and implementing the infrastructure. Using the product design based on the conceptual analysis, the product is developed to support data exchanges required to meet the objectives. Operating agreements that are not necessarily documented elsewhere need to be defined and established. The purpose of the implement designs activity is to complete whatever modifications are necessary to ensure the product satisfies the requirements of the conceptual model and product design, to produce and exchange data, and to abide by established agreements. The purpose of the implement the infrastructure activity is to implement, configure, and initialize the infrastructure necessary to support the execution and intercommunication of all components.
MIL-STD-499C	Not included

Table C-4: Product Development (continued)

INCOSE Handbook (v3)	In this standard, three steps make up the product development phase of the MSDBP SE process. The steps are implementation, integration, and transition to use. The purpose of the implementation step is to design, create or fabricate a system element conforming to that element's detailed description. The element is constructed employing appropriate technology and industry practices. During this phase, the requirements allocated to the system element to design, fabricate, code, or build each individual element using specified materials, processes, physical or logical arrangements, standards, technologies, and/or information flows outlined in detailed drawings or other design documentation are followed. Requirements are verified and stakeholder requirements are validated. If subsequent configuration audits reveal discrepancies, recursive interactions occur with predecessor activities or processes as required to correct them. The purpose of the integration step is to realize the system-of-interest by progressively combining system elements in accordance with the architectural design requirements and the integration strategy. This process is successively repeated in combination with the Verification and Validation Processes as appropriate. The integration step includes activities to acquire or design and build enabling systems needed to support the integration of system elements and demonstration of end-to-end operation. This step confirms all boundaries between system elements have been correctly identified and described, including physical, logical, and human-system interfaces; and confirms that all functional, performance, and design requirements and constraints are satisfied. Interim assembly configurations are tested to assure correct flow of information and data across interfaces to reduce risk, and minimize errors and time spent isolating and correcting them. The purpose of the transition to use step is to transfer custody of the system and responsibility for system support from one organizational entity to another. This includes (but is not limited to) transfer of custody from developers to users. This step installs a verified system in the operational environment along with relevant enabling systems. As part of this process, the user accepts that the system provides the specified capabilities in the intended operational environment. The transition to use step should be carefully planned to avoid surprises and recrimination on either side of the agreement; and transition plans should be tracked and monitored to ensure all activities are completed to both parties' satisfaction. Activities include the preparation of a transition to use strategy, user training, final confirmation that the system meets user needs, and documentation of post-implementation problems. At the conclusion of this step, the system is installed, acceptance criteria are met or discrepancies documented with recommended and agreed corrective actions.
CMMI-DEV	The Technical Solution process area also focuses on design implementation that is a component of the product development phase of the MSDBP SE process. Product components, and associated support documentation, are implemented from their designs. The other component of product development that is contained in this particular process document is product integration. The purpose of Product Integration is to assemble the product from the product components, ensure that the product, as integrated, functions properly, and deliver the product. A critical aspect of product integration is the management of internal and external interfaces of the products and product components to ensure compatibility among the interfaces. Product integration is more than just a one-time assembly of the product components at the conclusion of design and fabrication. Product integration can be conducted incrementally, using an iterative process of assembling product components, evaluating them, and then assembling more product components.
MSDBP	The purpose of this phase of the M&S development process is to build the M&S application defined by the product design. This primarily involves implementing a controlled software development process to implement the product design, although even standalone M&S applications could potentially have hardware-in-the-loop. This phase generally requires considerable iteration with the testing activities defined in the subsequent phase.

Table C-5: Product Testing

ANSI/EIA-632	For this particular standard, system verification process maps to the product testing technical process in the MSDBP SE framework. It is used to ascertain that: (1) the system design solution is consistent with its source requirements (selected preferred physical solution representation); (2) end products at each level of the system structure implementation, from the bottom up, meet their specified requirements; (3) enabling product development or procurement for each associated process is properly progressing; and (4) required enabling products will be ready and available when needed to perform. The three requirements associated with the system verification process are design solution verification, end product verification, and enabling product readiness. In the design solution verification step, the developer shall verify that each end product defined by the system design solution conforms to the requirements of the selected physical solution representation. In the end product verification step, the developer shall verify that an end product to be delivered to an acquirer conforms to its specified requirements. In the enabling product readiness step, the developer shall determine readiness of enabling products for development, production, test, deployment/installation, training, support/maintenance, and retirement or disposal.
ISO/IEC-15288	The purpose of the validation process is to provide objective evidence that the services provided by a system when in use comply with stakeholders' requirements, achieving its intended use in its intended operational environment. This process performs a comparative assessment and confirms that the stakeholders' requirements are correctly defined.
ISO/IEC-26702	In this particular standard, product testing is defined by the functional configuration audits and production approval reviews stages. Functional configuration audits should be completed to verify that products have achieved requirements; that they satisfy the characteristics as specified in specifications, interface specifications, and other baseline documentation; and the test plans and procedures were complied with. The production approval reviews should be completed after the audits to demonstrate that the total system has been verified to satisfy specification and baseline requirements.
IEEE 1516.3 (FEDEP)	The purpose of this step of the FEDEP is to plan the execution, established all required connectivity, and conduct testing prior to the execution. The main purpose of the plan the execution activity is to fully describe the execution environment and develop an execution plan. Additional activities include the incorporation of any necessary refinements to test and VV&A plans, and the development of a security test and evaluation plan, if necessary. Operation planning is also a key aspect to the activity. The purpose of the integration activity is to bring all of the participants into a unifying operating environment. This requires that all hardware and software assets are properly installed and interconnected in a configuration that can satisfy all data exchange requirements and agreements. Integration is normally performed in close coordination with testing and iterative "test-fix-test" approaches are used extensively. The purpose of the test activity is to ensure that the project operates to the degree required to achieve objectives. The desired output from this activity is an integrated, tested, validated, and if required, accredited project that indicates execution may commence.
MIL-STD-499C	Not included
INCOSE Handbook (v3)	In this standard, the verification phase maps to the product testing phase of the MSDBP SE process. The purpose of the verification step is to confirm that all requirements are fulfilled by the system elements and eventual system-of-interest, i.e., that the system has been built right. This step establishes the procedure for taking remedial actions in the event of non-conformance. The verification step confirms that all elements of the system-of-interest perform their intended functions and meet the performance requirements allocated to them. Verification methods include test, inspection, analysis, and demonstration. Verification activities are determined by the perceived risks, safety, and criticality of the element under consideration.

Table C-5: Product Testing (continued)

CMMI-DEV	In this particular process, the Verification process area maps to product testing in the MSDBP SE process. The purpose of Verification is to ensure that selected work products meet their specified requirements. The Verification process area involves the following: verification preparation, verification performance, and identification of corrective action. Verification includes verification of the product and intermediate work products against all selected requirements, including customer, product, and product component requirements. Verification is inherently an incremental process because it occurs throughout the development of the product and work products, beginning with verification of the requirements, progressing through the verification of the evolving work products, and culminating in the verification of the completed product.
MSDBP	The purpose of this phase of the M&S development process is to ensure that the developed M&S application meets all requirements and satisfies all stakeholder expectations. The output of this phase is the final product of the M&S development effort.

Table C-6: Project Management Practices

ANSI/EIA-632	The project management practices for this systems engineering process include systems analysis, requirements validation, system verification, and end product validation. The systems analysis process includes assessing alternative representations, conducting trade-off analysis, and performing risk analysis. The requirements validation process ensures that the requirements are necessary and sufficient for creating design solutions appropriate to meeting the exit criteria of the applicable engineering phase of project management practices and of the enterprise-based project management phase in which the engineering or reengineering efforts occur. The system verification process is used to ascertain that the product design (design solution) generated by implementing the specified requirements is consistent with its source requirements (physical solution representation); that the end products meet specified requirements; and that product development is progressing; and that required enabling products will be ready and available when needed. The end product validation is used to demonstrate that the products satisfy the validated requirements that were put into the system design process and that are applicable to the resulting end products.
ISO/IEC-15288	The project management practices for this particular standard include requirements analysis, risk management, configuration management, and verification and validation. The purpose of the requirements analysis process is to transform the stakeholder, requirement-driven view of desired services into a technical view of a required product that could deliver those services. This process builds a representation of a future system that will meet stakeholder requirements and that, as far as constraints permit, does not imply any specific implementation. It results in measureable system requirements that specify, from the supplier's perspective, what characteristics it is to possess and with what magnitude in order to satisfy stakeholder requirements. The purpose of the risk management process is to identify, analyze, and treat and monitor the risks continuously. The risk management process is a continuous process for systematically addressing risk throughout the evolution of a system product or service. The purpose of the configuration management process is to establish and maintain the integrity of all identified outputs of a project or process and make them available to concerned parties. The purpose of the verification process is to confirm that the specified design requirements are fulfilled by the system. This process provides the information required to effect the remedial actions that correct non-conformances in the system. The purpose of the validation process is to provide objective evidence that the services provided by a system when in use comply with stakeholders' requirements, achieving its intended use in its intended operational environment. This process performs a comparative assessment and confirms that the stakeholders' requirements are correctly defined.

Table C-6: Project Management Practices (continued)

ISO/IEC-26702	The production and support stages map to the project management practices and are applied to correct deficiencies discovered during production, assembly, integration, and acceptance testing of a product or system. In addition, these stages are applied to evolve the product or system to implement an incremental change, resolve product or service deficiencies, or to implement planned growth.
IEEE 1516.3 (FEDEP)	For this particular standard, project management practices include configuration management, risk management, verification and validation, and trade studies leading to project agreements. Configuration management is the application of technical and administrative direction and surveillance to identify and document the functional and physical characteristics of a model or simulation, control changes, and record and report change processing and implementation status. Risk management is the identification, assessment, and prioritization of risks followed by coordinated and economical application of resources to minimize, monitor, and control the probability and/or impact of unfortunate events. For the most part, methodologies consist of the following elements: • identify, characterize, and assess threats • assess the vulnerability of critical assets to specific threats • determine the risk (i.e., the expected consequences of specific types of attacks on specific assets) • identify ways to reduce those risks • prioritize risk reduction measures based on a strategy. The processes of Verification, Validation, and Accreditation (VV&A) are fundamental to the establishment of both capability and confidence. Effective use of M&S is driven by an understanding of what the M&S is capable of representing and how well it represents it. Formally defined: Verification is the process of determining that a model implementation and its data accurately represent the Developer's conceptual description and specifications. Validation is the process of determining the degree to which a model and its associated data accurately represent the real world from the perspective of the intended uses of the model. Accreditation is an official determination that a model is acceptable for a specific purpose. This official decision of an Accreditation Authority is the culmination of a confidence building process wherein evidence of model capability, accuracy, and usability is gathered, evaluated, and compared with model use requirements unique to the intended purpose. The methodology used to build confidence in M&S consists of four key steps: • Defining the Intended Use • Assessing M&S Risk and Maturity • Building a Basis of Confidence (V&V) • Assessing Level of Confidence (Accreditation) A trade study is the activity of a multidisciplinary team to identify the most balanced technical solutions among a set of proposed viable solutions. These viable solutions are judged by their satisfaction of a series of measures or cost functions. These measures describe the desirable characteristics of a solution. They may be conflicting or even mutually exclusive. Trade studies are commonly used to find the configuration that best meets conflicting performance requirements.
MIL-STD-499C	Not included

Table C-6: Project Management Practices (continued)

INCOSE Handbook (v3)	In this standard, several phases comprise the project management practices of the MSDBP SE process. They are quality management, risk management, configuration management, information management, requirements verification and validation, trade study and sensitivity analysis, and interoperability analysis. The purpose of the quality management step is to make visible the goals of the enterprise toward customer satisfaction. Enterprise policies and procedures govern the products, services, and implementations of the system project management practices to assure that they meet quality objectives and customer requirements. The quality management step establishes, implements, and continuously improves the focus on customer satisfaction and enterprise goals and objectives. There is a cost to managing quality as well as a benefit. The effort and time required to manage quality should not exceed the overall value gained from the process. Risk and opportunity management is a disciplined approach to dealing with uncertainty that is present throughout the entire evolution of the system. The objective is to achieve a proper balance between risk and opportunity. The activities in this step are used to understand and avoid the potential cost, schedule, and performance/technical risks to a system, and to take a proactive and structured approach to anticipate negative outcomes, respond to them if they occur; and to identify potential opportunities that may be hidden in the situation. The risk management activities include identifying and defining risk situations, analyzing risks for likelihood and severity in order to determine the magnitude of the risk and its priority for handling, defining the handling scheme and resources for each risk, using the criteria for acceptable and unacceptable risk, generate a plan of action when the risk threshold exceeds acceptable levels, maintaining a record of risk items and how they were handled, and maintaining transparent risk management communications. The objective of configuration management is to ensure effective management of the evolving configuration of a system, both hardware and software, during the product maturation. Fundamental to this objective is the establishment, control, and maintenance of software and hardware baselines. Baselines are reference points for maintaining development and control. These baselines, or reference points, are established by review and acceptance of requirements, design, and product specification documents. The primary output of the configuration management step is the maintenance of the configuration baseline for the system and system elements. Items are placed under formal control as part of the decision-making process. The required configuration baseline documentation is developed and approved in a timely manner to support required systems engineering technical reviews, the system's acquisition and support strategies, and production. This documentation is maintained throughout the life of the system. Configuration management formally documents the impact to any process, organizations, decisions, products, and services affected by a given change request. Information Management ensures that information is properly stored, maintained, secured, and accessible to those who need it thereby establishing/maintaining integrity of relevant system artifacts. Information management provides the basis for the management of and access to information throughout the evolution of the system, including after disposal if required. Designated information may include enterprise, project, agreement, technical, and user information. The mechanisms for maintaining historical knowledge in the prior processes - decision-making, risk and configuration management - are under the responsibility of information management. The output of this step is the availability for use and communication of all relevant systems artifacts in a timely, complete, valid and, if required, confidential manner. As part of the project management practices, requirements verification and validation are conducted. The verification step analyzes requirements for clarity, completeness and consistency, ensures that the requirements satisfy stakeholders' objectives, and achieve stakeholders' agreement. The validation step is to confirm that the requirements comply with stakeholders' needs. Another part of the project management practices for this standard is trade study and sensitivity analysis. Trade study describes a process for comparing the appropriateness of different technical solutions. The characteristics of each option are traded against each other.

Table C-6: Project Management Practices (continued)

	Once a best alternative has been identified, the stakeholders in the decision will want to know how sensitive the recommended selection is to differently evaluated criteria or to different estimates of the alternatives' characteristics - perhaps a different best alternative would result. Therefore, a good trade study provides a disciplined process that justifies the selected approach, and includes sensitivity analysis. A sensitivity analysis involves varying each utility and each weight and re-computing the weighted total for each alternative to ascertain what would change if the values of the utilities and weights were different. The significance of the change is best determined through conversations with stakeholders and subject matter experts. Interoperability analysis is part of the project management practices for this standard. Interoperability depends on the compatibility of components of a large and complex system to work as a single entity.
CMMI-DEV	In this particular process, the process area that make up project management practices include decision analysis and resolution, quality management, configuration management, risk management, and verification and validation. The purpose of Decision Analysis and Resolution is to analyze possible decisions using a formal evaluation process that evaluates identified alternatives against established criteria. The purpose of Quantitative Project Management is to quantitatively manage the project's defined process to achieve the project's established quality and process-performance objectives. The purpose of Configuration Management is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits. The purpose of Risk Management is to identify potential problems before they occur so that risk-handling activities can be planned and invoked as needed across the life of the product or project to mitigate adverse impacts on achieving objectives. The purpose of Validation is to demonstrate that a product or product component fulfills its intended use when placed in its intended environment. The purpose of Verification is to ensure that selected work products meet their specified requirements.
MSDBP	In order to conduct a successful project, there are many more considerations that must be effectively addressed in addition to the technical processes. Supporting project management practices are primarily management activities that overlay every aspect of the product development. While project managers are generally responsible for the conduct of such activities, M&S developers are full participants in ensuring that these activities are effectively assimilated into the normal day-to-day process of M&S development.

APPENDIX D: BEST PRACTICES APPENDIX B BIBLIOGRAPHY

The citations in this section were sources for the best practices in addition to the survey responses. While all of these references were reviewed, not all resulted in inputs to the final set of best practices.

[1] Atkinson, K. (2004). Applying Design Patterns to Modeling and Simulation. In Fall Simulation Interoperability Workshop Proceedings, 04F-SIW-096. Orlando, FL, September 2004.

[2] Balci, O. (2001). A methodology for certification of modeling and simulation applications. *ACM Transactions on Modeling and Computer Simulation, 11*, pp. 352-377.

[3] Bellman, K., & Landauer, C. (1998). Designing Models. In D. Cloud and L. Rainey, (Eds.) *Applied modeling and simulation: An integrated approach to development and operation*. New York, NY: McGraw-Hill.

[4] Bennett, B., Hillestad, R., & Long, G. (1998). Producing and Managing Data. In D. Cloud and L. Rainey, (Eds.), *Applied modeling and simulation: An integrated approach to development and operation*. New York, NY: McGraw-Hill.

[5] Blais, C. L., Goerger, N. C., Richmond, P., Gates, B., & Pace, M. (2005). Data Mapping and Ontology Design for Common Maneuver Networks, 05S-SIW-140. In Spring Simulation Interoperability Workshop Proceedings. San Diego, CA, April 2005.

[6] Brandt, K. L., Van Doren, P., & Carr, F. (2005). Modeling and Simulation for the Operational Warfighter: Operational Decision Factors, 05S-SIW-152. In Spring Simulation Interoperability Workshop Proceedings. San Diego, CA, April 2005.

[7] Buss, A. (2004). Simkit Analysis Workbench for Rapid Construction of Modeling and Simulation Components, 04F-SIW-020. In Fall Simulation Interoperability Workshop Proceedings. Arlington, VA, September 2004.

[8] Campos, J., Hull, G., & Mamaghani, F. (2004). TCRS – A Methodology and Tool Set for Specifying Data Content, 04S-SIW-127. In Spring Simulation Interoperability Workshop Proceedings. Arlington, VA, September 2004.

[9] Cerenzia, J. L., Haley, T. B., Scrudder, R., Lounsbury, D. M., & Goddard, R. P. (2005). Practical Experiences in Creating Components from Legacy Simulations. In The Interservice/Industry Training, Simulation, and Education Conference (I/ITSEC) Proceedings. Orlando, FL, 2005.

[10] Cloud, D. J., & Rainey, L. B. (Eds.). (1998). *Applied modeling and simulation: An integrated approach to development and operation*. New York, NY: McGraw-Hill.

[11] Committee on Modeling and Simulation Enhancements for 21st Century Modeling and Acquisition. (2002). *Modeling and simulation in manufacturing and defense acquisition: Pathways to success.* Washington D. C.: National Academy Press.

[12] Connors, S., Gauldin, J., & Smith, M. (2002). Closed-Loop, Simulation-Based, Systems Engineering Approach to Life Cycle Management of Defense Systems. *WSC*, vol.1, pp. 893-900, 2002 Winter Simulation Conference (WSC'02).

[13] Davis, D. M., Baer, G. D., & Gottschalk, T. D. (2004). 21st Century Simulation: Exploiting High Performance Computing and Data Analysis. In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2004.

[14] deSylva, M. J., Lutz, R. R., & Osborne, S. R. (2004). The Application of the DoDAF within the HLA Federation Development Process. 04F-SIW-015. Simulation Interoperability Workshop Proceedings. Orlando, FL, September 2004.

[15] Dobey, V. T. & Foley, P. G. (2004). System Contextual Development of Physical Environmental Representations, 04S-SIW-116. In Spring Simulation Interoperability Workshop Proceedings, Arlington, VA. 2004.

[16] Duck, A. J., Dunbar, R. T., & Ray, D. P., Maj. (2005). The Operational Environment and Implied Limitations on Modeling Information Operations (05S-SIW-072). In Spring Simulation Interoperability Workshop Proceedings, San Diego, CA, April 2005.

[17] Evans, D., Johnson, S., Stayonoff, T., & Wolff, G. (2004). Using Modeling and Simulation throughout the Entire System Life Cycle Including an Example of How the Methodology was Successfully Applied to OPNET Based Communication Architecture Models, 04F-SIW-008. In Fall Simulation Interoperability Workshop Proceedings, Orlando, FL, September 2004.

[18] Fisher, D. L., McDermott, T., Bryan, J. L., & Janes, M. A. (2004). State Consistency and Event Integrity Monitor, 04S-SIW-032. In Spring Simulation Interoperability Workshop Proceedings, Arlington, VA, September 2004.

[19] Fulmer, S. (2004). Using UML and XMI to Represent Model Metadata, 04F-SIW-082. In Fall Simulation Interoperability Workshop Proceedings, Orlando, FL, September 2004.

[20] Gerber, W. J. & Lacy, L. W. (2004). Behavior Composability Support Through Standardized Ontology Representations. In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2004.

[21] Goldiez, B. (2004). Model Pedigree: A Requirement for Composability, 04S-SIW-155. In Spring Simulation Interoperability Workshop Proceedings, Arlington, VA, September 2004.

[22] Harmon, S., Youngblood, S., & Metz, M. (2004). Building a Theater-Level Ground Combat Referent from SME Knowledge, 04S-SIW-126. In Spring Simulation Interoperability Workshop Proceedings, Arlington, VA, September 2004.

[23] Harrison, N., Gilbert, B., Jeffrey, A., Lestage, R., Lauzon, M., & Morin, A. (2005). KARMA: Materializing the Soul of Technologies into Models. In The

Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2005.

[24] Hembree, L. A., Kunitani, C., & White, M. E. (2004). Master Environmental Library and Secure Access, 04S-SIW-123. In Spring Simulation Interoperability Workshop Proceedings, Arlington, VA, September 2004.

[25] Hodson, D. D., Gehl, D. P., & Baldwin, R. O. (2006). Building Distributed Simulations Utilizing the EAAGLES Framework. In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2006.

[26] Hollenbach, J. W. (2004). NDIA Systems Engineering Division Modeling and Simulation Committee Study Task Report: M&S Support to the New DoD Acquisition Process.

[27] Jacobs, M. S. (2004). The Portable Source Initiative - An Industry Perspective. In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) Proceedings, Orlando, FL.

[28] Jones, R. M., Wallace, A. J., & Wessling, J. (2004). An Intelligent Synthetic Wingman for Army Rotary Wing Aircraft. In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2004.

[29] Karr, C. R. (2005). Conceptual Modeling in OneSAF Objective System (OOS). In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2005.

[30] Kasputis, S., Oswalt, I., McCay, R. and Barber, S. (2004). Semantic Descriptors of Models and Simulations, 04S-SIW-070. In Spring Simulation Interoperability Workshop Proceedings, Arlington, VA, September 2004.

[31] Keuning, M., & Gerretsen, A. (2008). Model Driven Development for Distributed Simulation Using SysML. In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2008.

[32] Kilby, M. and Esmonde, L. (2005). Distributed Scenario Generation for Multi-Modal Training, 05S-SIW-057. In Spring Simulation Interoperability Workshop Proceedings, San Diego, CA, April 2005.

[33] Lunceford, W. H. (2002, June 26). The Institutionalization of Modeling & Simulation or Making M&S All It Can Be. Army Model and Simulation Office. [Briefing]

[34] Lunceford, W. H. (2002). The Institutionalization of Modeling & Simulation or Making M&S All It Can Be. In 5th Simulation-Based Acquisition/Advanced Systems Engineering Environment Conference Proceedings, McLean, VA, 2002.

[35] MacSween, P., & Wainer, G. (2004). On the Construction of Complex Models Using Reusable Components, 04S-SIW-124. In Spring Simulation Interoperability Workshop Proceedings, Arlington, VA, September 2004.

[36] McLean, C. R., Lee, Y. T., & Jain, S. (2007). Integration of Incident Management Simulation-based Training Applications. In Fall Simulation Interoperability Workshop Proceedings, Orlando, FL, September 2007.

[37] Miller, D. D., Janett, A. C., Nakanishi, M. E., Salemann, L. J., Miller, T.W., Birkel, P. A., Hovanec, D., Gray, C., De La Cruz, J., & Kohler, T. (2004). Interchange and Interoperability - Modeling Environmental Data with the Common Data Model Framework. In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2004.

[38] Nance, R. E. & Arthur, J. D. (2006). Software Requirements Engineering: Exploring the Role in Simulation Model Development. In Proceedings of the 2006 Operational Research Society Simulation Workshop, pp. 117-127, Coventry, England.

[39] Naval Postgraduate School. (2008). *Acquisition manager's guide to modeling and simulation*. Monterey, CA.

[40] Őren, T. I. (2005). Toward the Body of Knowledge of Modeling and Simulation (M&SBOK). In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2005.

[41] Őren, T. I. and Yilmaz, L. (2006). Synergy of Systems Engineering and Modeling and Simulation. In Proceedings of the 2006 International Conference on Modeling and Simulation - Methodology, Tools, Software Applications, pp. 10-17. Calgary, AL, Canada. July 31 - August 2, 2006.

[42] Pace, D. K. (2000). "Conceptual Model Development for C4ISR Simulations," 5th International Command and Control Research and Technology Symposium. In Proceedings of the 5th International Command and Control Research and Technology Symposium, Australia War Memorial, Canberra, Australia, October 24-26, 2000.

[43] Pace, D. K. (2009). Survey response.

[44] Parsons, D. (2009). Survey response.

[45] Pournelle, P. E., LCDR, Blais, C. L., & Brutzman, D. (2004). A Transformational Framework for Design, Development, and Integration of Simulation Models, 04F-SIW-019. In Fall Simulation Interoperability Workshop Proceedings, Orlando, FL, September 2004.

[46] Pratt, D.R. & Franceschini, R.W. (2006). "Training" for Models: The Role of Knowledge Management in Applying Modeling and Simulation (M&S) to Systems Engineering. In Proceedings of the 9th NDIA Systems Engineering Conference, San Diego, CA, 2006.

[47] Richey, F. (2002). Modeling and Simulation CMMI: A Conceptual View. *CrossTalk: The Journal of Defense Software Engineering, 15*(3), 29-30.

[48] Robinson, S. (2006). Conceptual Modeling for Simulation: Issues and Research Requirements. In WSC '06: Proceedings of the 38th Winter Simulation Conference, pp. 792-800. Winter Simulation Conference, Monterey, CA, 2006.

[49] Robinson, S. (2004). *Simulation: The practice of model development and use*. Hoboken, NJ: John Wiley & Sons, Ltd.

[50] Roman, P.A. (2005). Garbage In, Hollywood Out! In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2005.

[51] Schloman, J. (2005). A Logical Data Model and Translation Software for Scenario Representations in Mission-Level Simulations, 05S-SIW-160. In Spring Simulation Interoperability Workshop Proceedings, San Diego, CA, April 2005.

[52] Schuette, R. & Rotthowe, T. (1998). The Guidelines of Modeling - An Approach to Enhance the Quality in Information Models. In T. W. Ling, S. Ram, & M. Lee (Eds.) Conceptual Modeling - ER '98: 17th International Conference on Conceptual Modeling, Proceedings, pp. 240-254. Singapore: Springer. November 16-19, 1998.

[53] Scrudder, R., Graves, W. H., Tiegen, T., Johnson, C., Hix, S., & Hollenbach, J. W. (2004). Improving Information Quality and Consistency for Modeling and Simulation Activities. In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2004.

[54] Shao, G. & Lee, Y. T. (2007). Applying Software Product Line Technology to Simulation Modeling of Emergency Response Facility. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology, 4* (4).

[55] Snyder, M. (2005). Ubiquitous 3D for the Network-centric Battlefield - Smaller is Better, 05S-SIW-125. In Spring Simulation Interoperability Workshop Proceedings, San Diego, CA, April 2005.

[56] Stytz, M. R. & Banks, S. B. (2004). Issues in Software Protection for Military Simulation. 04S-SIW-015. In Spring Simulation Interoperability Workshop Proceedings, Arlington, VA.

[57] Swain, J. J. (2003). Simulation reloaded. *OR/MS Today, 30*(4).

[58] Trbovich, S. & Reading, R. (2005). Simulation and Software Development for Capabilities Based Warfare: An Analysis of Harmonized Systems Engineering Processes, 05S-SIW-106. In Spring Simulation Interoperability Workshop Proceedings, San Diego, CA, April 2005.

[59] Valle, T. & McGregor, B. (2004). The DMT Master Conceptual Model. In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2004.

[60] Wainer, G. & MacSween, P. (2004). On the Construction of Complex Models Using Resuable Components, 04S-SIW-124. In Spring Simulation Interoperability Workshop Proceedings, Arlington, VA, September 2004.

[61] Wall, J., Elms, R., & Nock, D. (2004). Developing an Incident Management Simulation for Training Emergency Responders. In The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2004.

[62] Wang, X., Zhang, L., & Huang, K. (2004). Improving Simulation Conceptual Model, 04F-SIW-036. In Fall Simulation Interoperability Workshop Proceedings, Orlando, FL, September 2004.

[63] Watkins, J. C., Hare, C., Scrudder, R. O., Landrum, O. & Busbee, M. (2004). Composability Perspectives within the Threat Modeling and Analysis Program (TMAP). In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2004.

[64] Wolff, G., Johnson, S., Evans, D., & Stayonoff, T. (2004). Using Modeling and Simulation throughout the Entire System Life Cycle, 04F-SIW-008. In Fall Simulation Interoperability Workshop Proceedings, Orlando, FL, 2004.

[65] Yao, K., Wagenbreth, G., & Ward, C.E. (2006). Agile Data Logging and Analysis. In The Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) Proceedings, Orlando, FL, 2006.

APPENDIX E: ABBREVIATIONS AND ACRONYMS

AMSMP	Acquisition M&S Master Plan
AMSWG	Acquisition Modeling and Simulation Working Group
ANSI	American National Standards Institute
CJCSI	Chairman of the Joint Chiefs of Staff Instruction
CM	Conceptual Model
CMMI	Capability Maturity Model Integration
CMMI-DEV	CMMI for Development
CMMI-SE/SW	CMMI for Systems Engineering and Software Engineering
COSO	Committee of Sponsoring Organization
DIS	Distributed Interactive Simulation
DMSO	Defense Modeling and Simulation Office (now M&S CO)
DoD	Department of Defense
DoDAF	Department of Defense Architecture Framework
EIA	Electronic Industries Alliance
EIR	Engineering Integration Review
FAIT	Fabrication, Assembly, Integration, and Test
FEDEP	Federation Development and Execution Process
FOM	Federation Object Model
FoS	Family of Systems
GUI	Graphical User Interface
HLA	High Level Architecture
HW	Hardware
I/ITSEC	Interservice/Industry Training, Simulation, and Education Conference
I/O	Input/Output
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
INCOSE	International Council on Systems Engineering
IPPD	Integrated Product and Process Development
ISO	International Organization for Standardization
JCIDS	Joint Capabilities Integration and Development System
JHU/APL	The Johns Hopkins University Applied Physics Laboratory
JTC	Joint Technical Committee
KA/KE	Knowledge Acquisition/Knowledge Engineering

M&S	Modeling and Simulation
M&S	Models and Simulations
M&S CO	Modeling and Simulation Coordination Office
MDA	Model Driven Architecture
MIL-STD	Military Standard
MSDBP	Models and Simulations Development Best Practices
MWBP	Mobile Web Best Practices
NDIA	National Defense Industrial Association
NSSAP	National Security Space Acquisition Policy
OMG	Object Modeling Group
OSI	Open Systems Interconnections
POC	Point of Contact
QA	Quality Assurance
RUP	Rational Unified Process
SE	Systems Engineering
SEDRIS	Synthetic Environment Data Representation and Interchange Specification
SEP	Systems Engineering Process
SG	Study Group
SISO	Simulation Interoperability Standards Organization
SIW	Simulation Interoperability Workshop
SME	Subject Matter Expert
SoS	Systems of Systems
STEP	Standard for the Exchange of Product model data
SW	Software
SysML	Systems Modeling Language
ToR	Terms of Reference
UML	Unified Modeling Language
V&V	Verification and Validation
VV&A	Verification, Validation, and Accreditation
WBS	Work Breakdown Structure
WCAG	Web Content Accessibility Guidelines
XMI	XML Metadata Interchange
XML	Extensible Mark-up Language
XSLT	Extensible Stylesheet Language Transformation

NATIONAL SECURITY ANALYSIS DEPARTMENT
THE JOHNS HOPKINS UNIVERSITY • APPLIED PHYSICS LABORATORY
Johns Hopkins Road, Laurel, Maryland 20723-6099