Enhancing Systems Engineering Agility with Modeling and Simulation

Timothy M. Frey and Matthew C. Valencia

In the face of globalization, with the rapid pace at which the very nature of warfighting is changing, and with ever-increasing rates of technological innovation, much attention has been given in recent years to transforming traditional systems engineering and standard acquisition paradigms to meet these challenges. We argue that, by applying principles from agile software development, which has achieved strong success in recent years, there is great potential to meet these challenges directly and, in doing so, to save money, increase efficiency, and ensure that the right decisions are being made as systems are developed and deployed. Furthermore, we suggest that this movement to "agile systems engineering" can largely be accomplished by employing systems engineering practices that are centered on evolutionary, end-to-end implementations of physics-based modeling and simulation. Finally, we argue that the DoD must have qualified, independent, trusted agents that will help it execute these improved systems engineering practices if it is to be successful.

INTRODUCTION

The Global Engagement Department (GED) continues to strive to help the DoD address its future needs. In both the core, legacy GED programs in the Precision Engagement and Strategic Systems Business Areas (Tomahawk and Trident, respectively) as well as for newer programs, GED has been challenged by its sponsors to define how it can best support them in current system modification and support activities as well as in new systems development. These challenges reflect a broader need in the DoD to better understand and navigate the future in the face of globalization, dynamic adversaries, and persistent competition for acquisition dollars. To meet these challenges, we must find ways to significantly improve the way that systems engineering processes often are executed. The push for faster development times, lower costs, longer shelf lives, and more flexible deployment options, as well as the recognition of the DoD's trend toward even more complex and more interdependent programs in the midst of already inadequate acquisition processes, almost mandates that this transformation occur.[1]

We argue that by borrowing and applying concepts from agile software development, which has achieved notable success in recent years, there is great potential to transform traditional systems engineering processes to save money, increase efficiency, and ensure that the right decisions are being made as systems are developed and deployed. As this thesis is not entirely new,[2] we move beyond it to suggest that this transformation to "agile systems engineering" can be largely accomplished by employing systems engineering practices that are grounded in an evolutionary, end-to-end utilization of physics-based modeling and simulation (M&S). In addition to software engineering, agile approaches have been successfully applied to other domains (e.g., project management[3]), and we believe similar success could be realized in applying the fundamental values and principles of the approach to systems engineering, primarily through physics-based M&S.

Finally, we suggest that the DoD should have trusted agents that would help it to implement this transformation. This role is well-suited to APL's unique culture and organizational "mantras." Recent interactions with several of our sponsors suggest that they are looking to us to help them, independently, with their future systems engineering challenges. These sentiments reinforce the need for us to think hard about how we can continue to leverage our unique organizational identity, history of strong performance, and vision for the future to help them in this strategic role.

THE TRANSFORMATION TO AGILE SYSTEMS ENGINEERING

Traditional Systems Engineering

Wikipedia's definition of systems engineering is "an interdisciplinary field of engineering that focuses on how complex engineering projects should be designed and managed."[4] Its formalization and broad-scale usage began during the 1940s,[5] and ever since it has been recognized as necessary for the successful creation of systems, especially for those that are complex. Most standard views of the systems engineering process revolve around a progression from the identification of requirements through concept/capability assessment and exploration to solution validation, implementation, and deployment. Figure 1 shows the systems engineering "V" diagram, which often is used to illustrate this concept.

Although the basic construct of this process is sound, several potentially undesirable characteristics are present in many traditional implementations:

- There is a strong emphasis on defining requirements up front in their entirety and usually a strong resistance to changing them as the system is developed.[6] A relatively sequential process typically is used to progress through systems engineering "phases." Even if some phases are worked in parallel, once they are complete, cost and schedule pressures make it difficult to revisit them.[6]
- Many times, for systems that involve hardware, hardware prototypes are built early and often to explore concepts and define capability (although this is happening less frequently as software technologies increase in capability). This practice leads to the absorption of high costs of change early in system development.
- For the DoD, there typically is a tight coupling between the systems engineering and acquisition processes,[7,8] which results in more resistance to change and, most likely, less efficiency.

We recognize that these characteristics do not hold universally true for all implementations of the systems engineering process; nevertheless, we believe that they are common enough that we will contrast them with the changes that we are recommending. Furthermore, we do not advocate completely abandoning traditional systems engineering techniques; instead, we suggest augmenting (and, where appropriate, removing) those techniques that limit efficiency.

The Need for Change: Borrowing from Agile

The challenges to systems development in today's environment, outlined above, drive us to an approach that is different from what has been employed in the past. Just as the software engineering community has faced analogous challenges and largely addressed them through agile software development, we believe that using a similar approach may yield equally dramatic improvements in systems engineering. These values and principles are defined in detail in *The Manifesto for Agile Software Development*[9] and are summarized below:

- Strong focus on customer satisfaction and continuous customer involvement in product development
- Continuous integration and frequent delivery of incrementally useful products (weeks versus months)

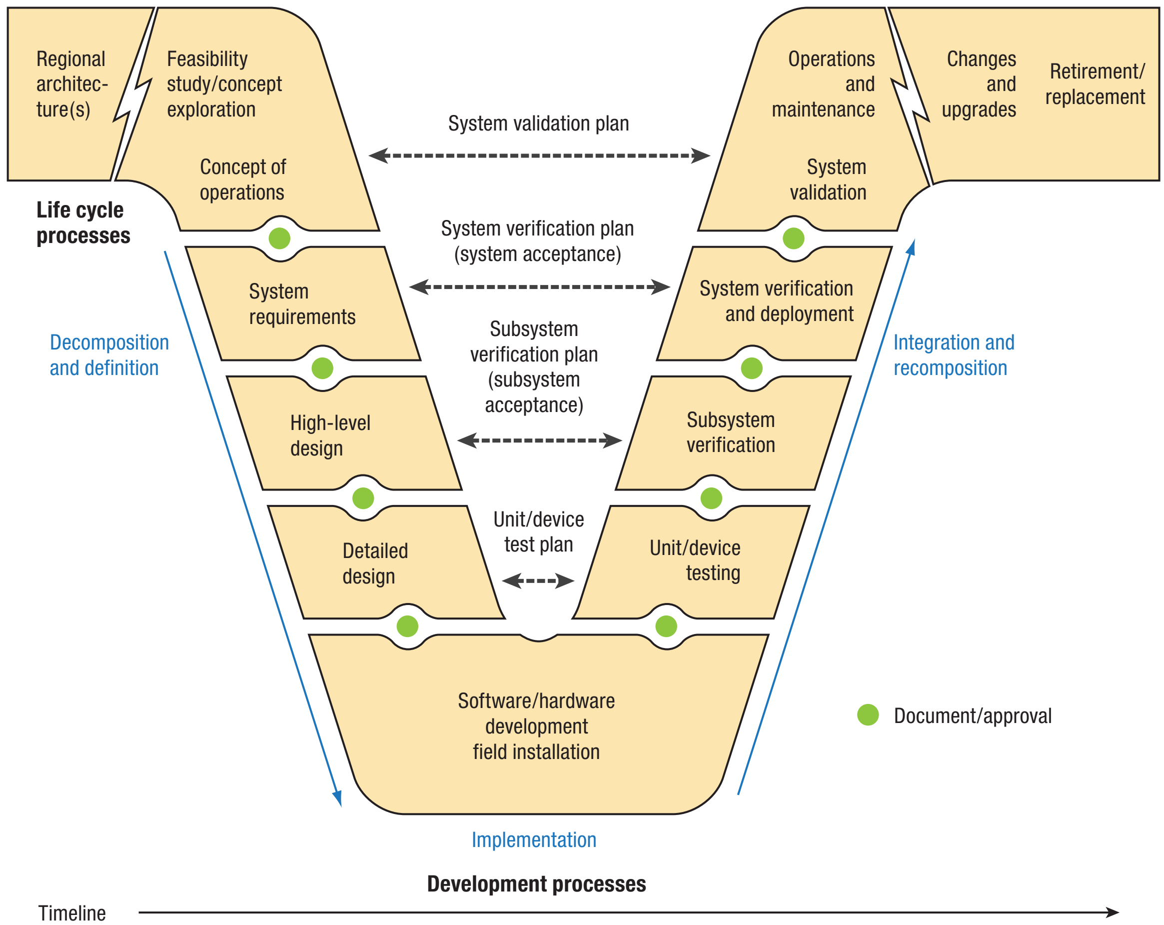

Figure 1. The systems engineering "V" diagram.[19]

- Requirements changes embraced, and managed, throughout the development cycle
- Development performed by self-organizing, motivated teams with a strong focus on face-to-face conversation, close cooperation, and trust
- Progress primarily measured through *functional* product updates achieved through *test-driven* development

Many questions arise in this proposed leap from software development to systems engineering. The most serious of these involve the development and integration of hardware, as hardware is the primary difference between software engineering and systems development (many software applications are, in fact, software "systems" that have to perform required functions to established standards with some level of accuracy and reliability, just like any other system). The inclusion of hardware accelerates the increase of the cost of change as the system is developed, and, in fact, traditional systems engineering implementations (and older software development techniques) were designed to mitigate this effect by spending as much time up front defining requirements so that changes to those requirements could be avoided as much as possible during development. This "change avoidance" has led to several of the disadvantages of traditional systems engineering implementations mentioned above.

As the potential benefits of applying agile concepts to systems engineering already are beginning to be explored in the literature,[2] we present these ideas simply as an introduction for the reader to agile principles, and we accept that increased agility in systems engineering will bring tangible benefits. We base this conclusion on a variety of sources from both systems engineering and system acquisition perspectives, not the least of which is a report from the Government Accountability Office that states "the conventional acquisition process is not agile enough for today's demands."[1] For the remainder of the article, we focus instead on the best method to enable this transition to agile systems engineering, which in our opinion, can best be achieved through an end-to-end, evolutionary implementation of physics-based M&S.

Using M&S to Enable the Transformation

History of Using M&S for Systems Engineering

The idea that M&S can benefit systems engineering has existed for several decades[6]; one of the first widely cited uses of M&S was in the modeling of nuclear detonation during the Manhattan Project.[10] In the past two decades, as software technology has improved, the number of documented cases of M&S improving systems engineering activities has increased. Although encouraging, many of these efforts have been disjointed and suffered from inefficient redundancies and poor management. In 1997, the Acquisition Council of the DoD Executive Council for Modeling and Simulation adopted a broader vision for simulation-based acquisition (SBA).[11–13] The vision of SBA was to enable the "collaborative use of simulation technology that is integrated across acquisition phases and programs."[14] Although SBA promised to yield increased cost effectiveness and efficiency early on in system development, its vision was never fully realized. A Pentagon-funded survey,[15] in addition to other reports, found the following:

- The limited tenure of program managers, coupled with the demand for immediate results, produces no incentive to invest in long-term, expensive M&S
- There often is duplication of effort that results from programs spending money on similar M&S components.

In addition to these problems, we can identify other issues:

- SBA has some positive elements and has had limited success; however, it does not include an overall process that addresses the roles of various supporting organizations, nor does it include a systematic process to manage/track its execution even within organizations.[14]
- SBA does not cover system development from "cradle to grave."

Although remnants of SBA still exist, even the term itself has become fraught with conflict because of SBA's poor track record. Some have even gone so far as to label SBA a "myth."[15]

M&S-Based Systems Engineering: Overall Vision

Although SBA and other historical M&S efforts have made some inroads into improving systems engineering, the vision presented in this article proposes a complete and coordinated integration of M&S throughout the entire systems engineering process. The later section *M&S-Based Systems Engineering Process* contains a more specific discussion of implementation, but the basic overall concept is presented below.

Figure 2 shows a slightly different representation (than the one in Fig. 1) of the traditional systems engineering process (upper diagram). Although we do not take exception to any of the components of this process (critical needs, capability assessment, etc.), the implementation of it typically involves an early lock-down of requirements and early development of hardware prototypes/components that incur high costs when changes are required, as mentioned above. We believe that this early "requirements lock" is primarily because of the pressure placed on programs to show tangible progress as soon as possible, the tight coupling between

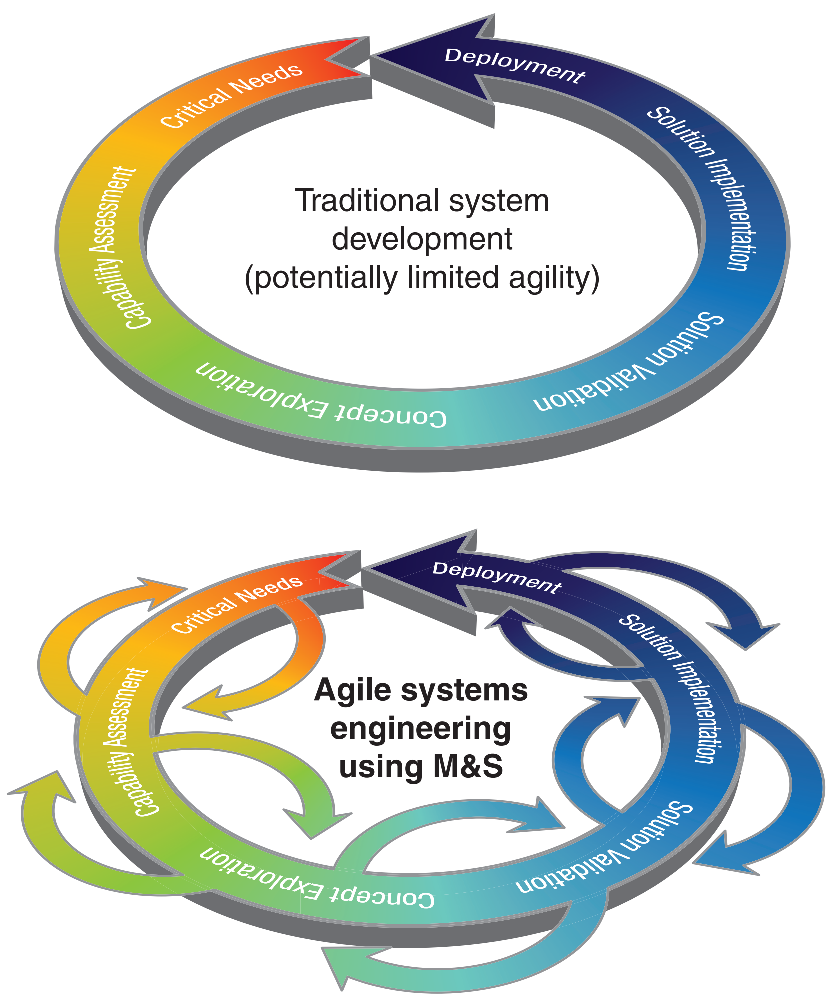

Figure 2. Agility in systems development.

systems engineering and system acquisition processes, and system developers who may not believe that they can answer relevant system design questions any other way. However, if representative, physics-based M&S is used instead to supplement development in a well-managed framework (represented by the lower diagram in Fig. 2), many, if not most, systems engineering process tasks can be performed virtually—exposing issues likely to be encountered during later stages of the process and providing an environment to explore and identify solutions wherein there is a low cost associated with making changes. Feedback from this "look ahead" approach (represented by the arrows in Fig. 2) can better inform efforts at current stages of system development and provide insights that will increase confidence that the right decisions are being made along the way. This approach is similar to the intent of agile software development because it enables a much wider look at requirements, options, performance envelopes, subsystem interactions, etc., by permitting iterative improvement at low change costs. In essence, the more one can do before "metal is bent" (and, indeed, *while* it is being bent), the more efficient, flexible, and powerful the process will be. Examples of the potential uses of M&S at each phase of the systems engineering life cycle are shown in Fig. 3.

In addition, with the remarkable progress in computer technology in the last decade, high-fidelity physics-based subsystem, system, and environment models are feasible—models that capture not only physics-based phenomena in real time or near real time but also emergent interaction effects for the complex environments in which these systems will be deployed. Advances in processor speed, computer memory, new hardware [e.g., high-performance graphics processing units (GPUs)], and algorithms to optimally utilize that hardware have been essential. Fast, physics-based approaches have been developed and used for special effects in the gaming and film industries and show promise for application to systems engineering (e.g., smoothed particle hydrodynamics). Finally, recent paradigm shifts in software development approaches (e.g., agile software development, as discussed in this article) have enabled drastic increases in efficiency and overall flexibility.[16]

M&S-BASED SYSTEMS ENGINEERING PROCESS

Process Description

Given that systems engineering will benefit from agile practices, and that these practices can be enabled by evolutionary, end-to-end use of physics-based M&S, the following sections outline a proposed high-level process with which to accomplish this transition. A representative flow diagram is shown in Fig. 4.

As motivated by Fig. 3, every phase of systems engineering presents a unique set of problems to be solved or questions to be answered. These "needs" could range from concept design and initial feasibility at the beginning of system development to mission planning and training near the end. The idea is that this set of needs (represented by the leftmost box in Fig. 4), in the context of the appropriate concept of operations (CONOPS), should be evaluated within an analysis framework. By using this framework, the types of data that need to be studied can be determined, appropriate metrics can be defined, the respective roles of M&S and field tests can be defined, and the paths that will most efficiently and

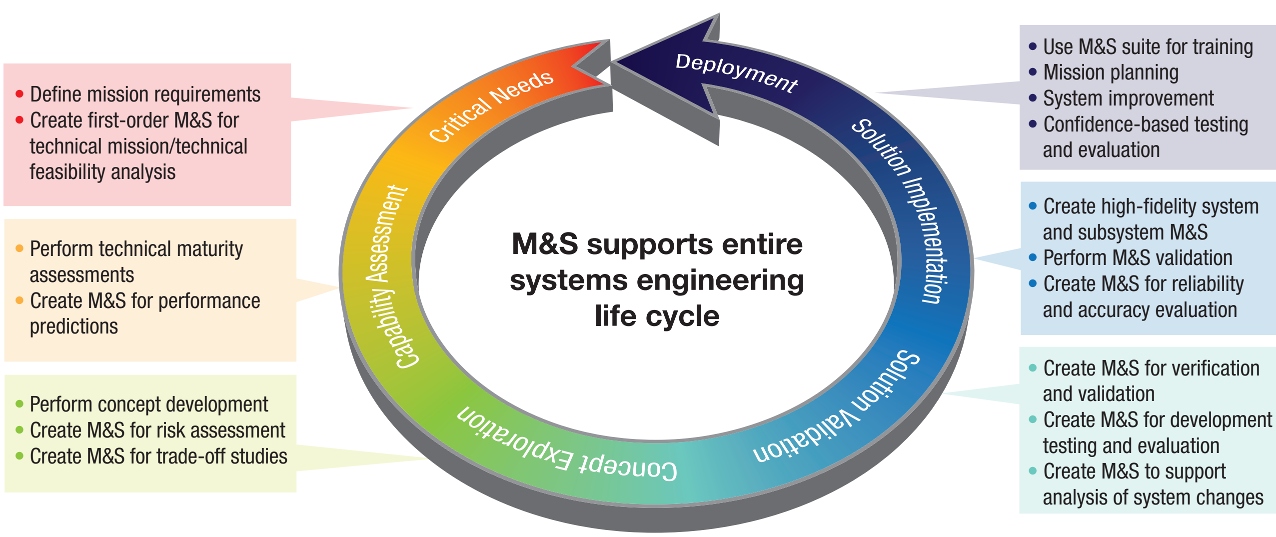

Figure 3. M&S supports the systems engineering life cycle.

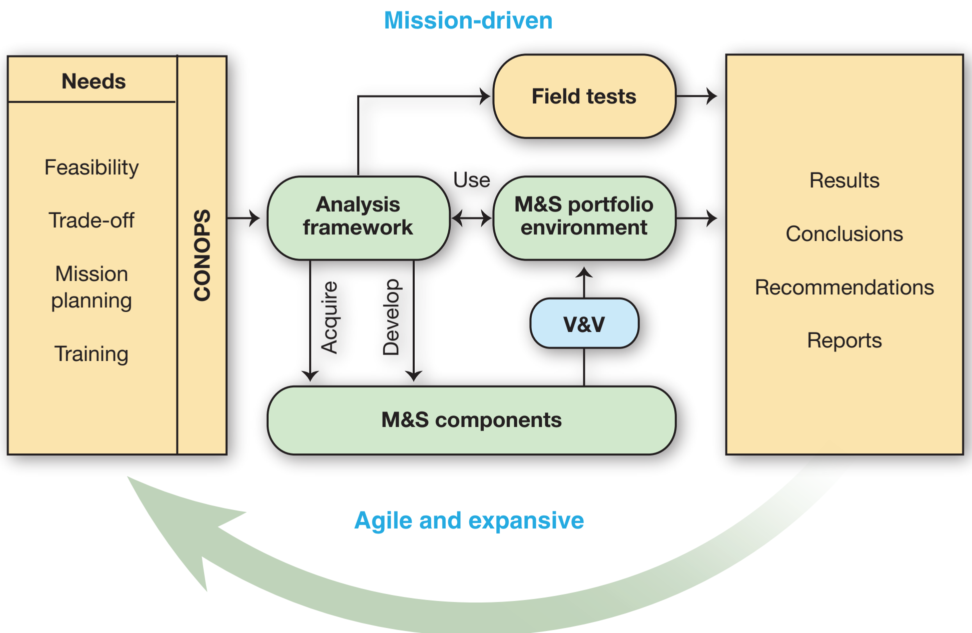

Figure 4. Notional M&S-based systems engineering process.

most cost-effectively address the initial needs can be decided. These actions will lead to results and conclusions, which will then initiate a new set of needs, and the process can iterate as necessary.

The M&S portfolio shown in Fig. 4 represents the set of extant M&S relevant to the system being developed at a given point in system development. It could be that some M&S components already exist and are available, and, if so, they may be used directly. If they do not exist or are not directly available, then they need to be acquired or developed and then must be verified and validated before integration into the M&S portfolio for use. The M&S portfolio, as determined by the analysis framework, will either address the needs sufficiently or cue up the appropriate field tests to “fill in the gaps.” By using this integrated process, field tests can be customized and designed to yield the maximum possible benefit. In some cases, M&S may help obviate field tests, thereby saving time and money. In other cases, M&S may show field tests to be indispensable, thereby ensuring that the system is properly scrutinized and provides the necessary reliability and performance.

Use of a structured, M&S-driven process can yield benefits within a given phase of systems engineering by yielding more agility; however, we believe that the true power in the concept presented in this article is the use of this approach across the entire systems engineering life cycle. The implementation of each process component will depend on the phase (specific needs, types of M&S used, etc.; see Fig. 3), but the overall process structure can be maintained. Specifically, as system development matures, the M&S portfolio will expand in size and capability as components are added. As a result, the portfolio will become increasingly capable of addressing systems engineering needs “out of the box,” thus reducing cost and improving analysis speed as the system is developed. As the size of the portfolio increases, the rate of M&S component addition should decrease, and the number of ways the portfolio can be used to perform systems engineering studies should increase. Of course, the M&S components must be interoperable to fully realize this capability improvement, but if the portfolio is managed by capable leadership (see *The Need for an M&S Portfolio Manager*), it certainly is possible. This iterative, overall process is shown in Fig. 5.

If successful, this evolutionary M&S portfolio expansion may, by design, provide a majority of the components necessary for efforts that typically are not conducted until the end of system development—efforts such as training, mission planning, and testing and evaluation. For example, models built during concept exploration to study the feasibility of a given sensor could be leveraged for integration in a mission planner for a system that contains that sensor. Simulations built to help validate subsystem performance in operational environments should be directly expandable into simulations capable of assessing the performance of the entire system. Models designed to capture subsystem interactions for use during system design should be usable for system evaluation and operator training. Historically, this investment in M&S reusability has not been fully realized as a life-cycle companion to systems engineering.

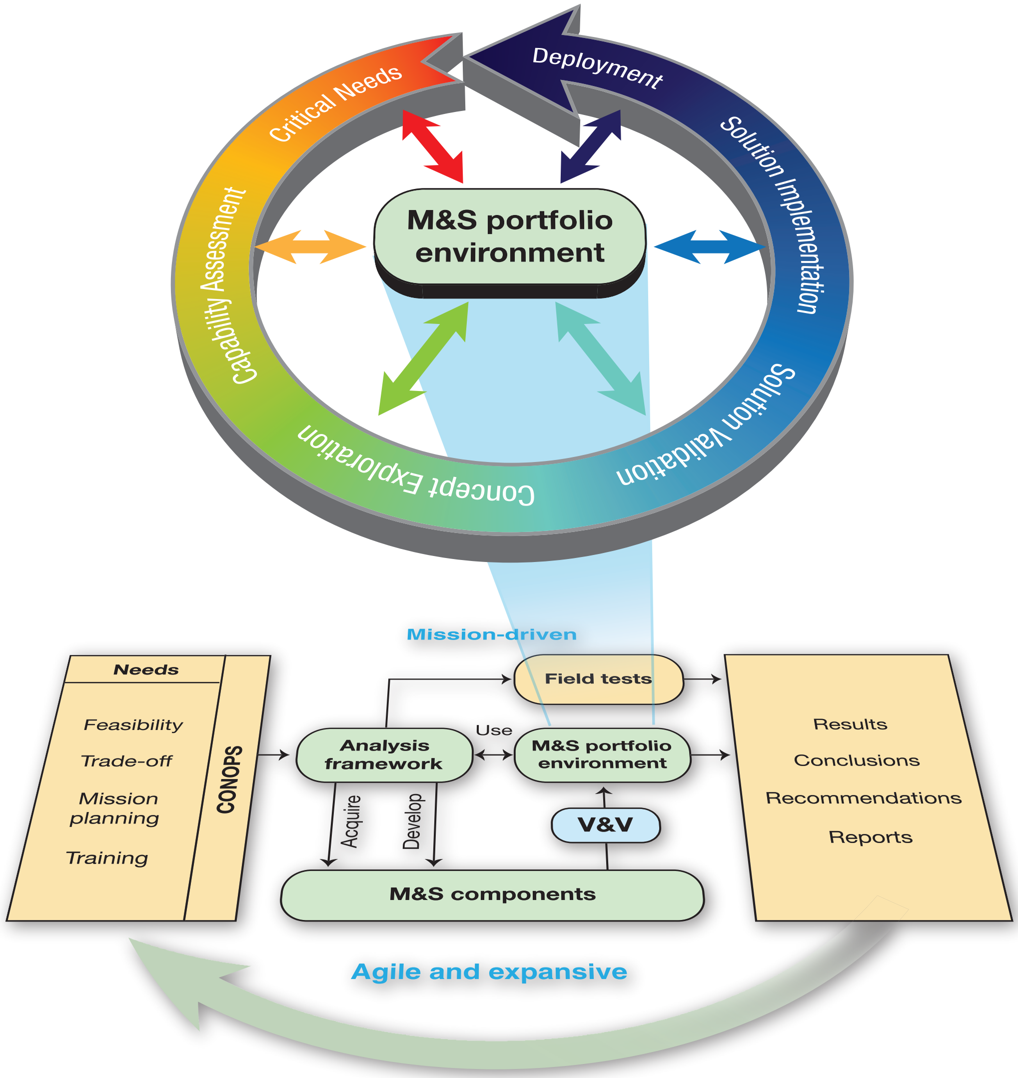

Figure 5. Process framework used at each stage.

Tenets of M&S Portfolio Implementation

The process described above is a start, but given the diversity of systems, and the many uncertainties at the beginning of system development, there is much to consider. Much like the agile software development community has developed a core set of values and principles, we identify a core set of tenets below that should be considered as a basis for process implementation.

> ***Tenet 1***
>
> Ensure that the evolving portfolio mirrors the real system's architecture and interactions. (*What system do you want to build?*)

If the abstraction between the real system and the M&S portfolio can be minimized, the M&S should be able to readily capture emergent behavior and identify unanticipated downstream consequences of system changes, upgrades, failures, etc. Without such M&S, these system interactions may not ever be adequately understood, even if they could be identified. M&S that mirrors the real system's architecture and subsystem interactions reduces the risk that significant modification will need to be made to model those interactions. This approach emphasizes developing integrated families of solutions to avoid building and maintaining a myriad of different point solutions.

> ***Tenet 2***
>
> Use an agile approach for developing the M&S portfolio. (*How are you going to build the system?*)

We have described above the thesis that systems engineering can be made more agile by evolutionary, end-to-end use of M&S; however, this tenet suggests that the creation and sustainment of the M&S portfolio itself be executed in an agile way. This paradigm focuses on continually maximizing value by building what is necessary when it is needed and allowing the M&S components to change (performance, scope, fidelity, etc.) as the systems engineering process is executed. This approach enables one to derive short-/shorter-term value from the portfolio, thus avoiding the traditional approach to investing in M&S, which typically gives only long-term returns (see *History of Using M&S for Systems Engineering*).

Tenet 3

Govern the development, verification and validation (V&V), and integration of each M&S component into the M&S portfolio. (*How are you going to manage development of the system?*)

Although agility enables flexibility while delivering constant value, it does not directly ensure that the portfolio evolves into a correct, normalized, and interoperable representation of the system—an issue also present in the development of a service-oriented architecture (SOA).[17] Best practices from the SOA community suggest that proper governance is critical to maintaining the quality and vision of the portfolio, especially with the participation of multiple organizations. Specific practices would likely differ depending on the system under development, but, in general, governance would consist of developing "an enforceable set of policies for building, deploying, [integrating], and managing [components]."[18] In fact, problems with organization, governance, coordination, and overall strategy are cited as one of the major reasons that SBA has had such limited success.[15] This begs the question of how we think our concept will work when SBA failed in this regard. In the best-case scenario, sponsors will include the development and integration of M&S into their contractual relationships with system developers, such that the M&S become formal deliverables in some way. (Governance is much easier when funding is at stake.) The specifics of how this would work will depend on the system and organizations involved. Even if this tenet is not fully realized, however, we do believe that something is better than nothing. At a minimum, if we can achieve partial buy-in from the organizations involved, it will be an improvement over the primarily stovepiped implementations currently in practice.

Tenet 4

Continually assess the relative cost versus benefit of using M&S throughout system development and deployment. (*When and how are you going to use the portfolio during system development?*)

We are suggesting that M&S should typically be the starting point for most systems engineering activities. However, at some point in a given systems engineering activity, there may be a cost-effective alternative (small-scale prototyping, actual testing, etc.) for deriving the answers needed to continue that particular activity. M&S should help with the design and structure of the alternatives (e.g., field tests) to maximize invested time and money, but consideration needs to be given to *not* using M&S when field tests and/or subsystem tests may be more efficient or effective. Figure 4 helps to illustrate this point.

THE NEED FOR AN M&S PORTFOLIO MANAGER

Concept

So far we have suggested that agile systems engineering could lead to significant improvements in efficiency, and we have introduced evolutionary, end-to-end M&S as a key enabler of this transformation. We have proposed a high-level process and listed a set of tenets that should followed to maximize the probability of success. However, without strong leadership and governance, there is no guarantee that the process will be followed, that the tenets will be considered, or that the overall strategy will be defined and implemented in a consistent, optimal way throughout system development.

Typically for system development, the prime contractor will subcontract various components of the system. The main responsibility of the subcontractors is to deliver a hardware and/or software product that meets the requirements of the contract, and, in turn, the prime contractor (or one of its surrogates) will integrate these components into the overall system. There is nothing wrong with this division of labor per se; however, there is a great deal of cost and time associated with change (e.g., if something does not work or a better solution is found somewhere along the way), especially because contract incentives, instead of evolving needs, can drive development.[7] In addition, there is no easy way for subcontractors to test their individual components at a system or mission level before "final" delivery. Most of the time, all they can do is build according to prespecified requirements, which often change during system development.

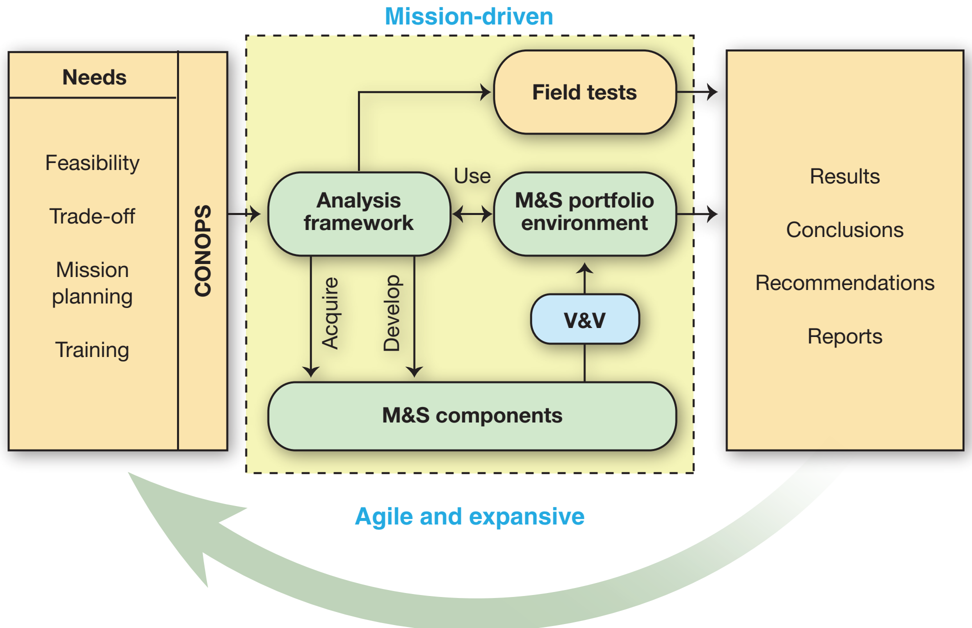

Figure 6. Role of the portfolio manager.

We believe that this dilemma can be at least partially addressed by the approach outlined in this article. The remaining problem is that, although the subcontractors may very well have used M&S as part of their development, they probably did so in the absence of an overall strategy for system M&S development,[8] including specific requirements for the interoperability and performance of M&S components. The end result is a plethora of (possibly redundant) M&S components related to the system that cannot be used together to answer any questions about the overall system itself. Their utility is bound to the system component for which they were built, and even worse is the reality that most of them cannot be used outside of contractor oversight because of intellectual property considerations. Intellectual property is important to protect, but there are ways to package proprietary components to ensure that they interoperate with the rest of the M&S portfolio but still do not give away company secrets (through dynamically linked libraries, web services, etc.). One survey of 22 major acquisition programs reveals that most of the M&S developed for specific projects within a program were unique to that project and owned by the contractors, not the government.[15] As a solution to these issues, we suggest an M&S portfolio manager, an organization empowered to manage a versatile and powerful plug-and-play M&S portfolio by defining performance and interoperability requirements and managing M&S development throughout system maturation (without necessarily needing the source code for all individual M&S components).

The areas of primary concern to the portfolio manager are highlighted in the yellow dashed box in the center of Fig. 6 (shown in the context of the process shown in previous figures). The portfolio manager should have some level of cognizance over the entire process as well as special responsibility to coordinate the activities highlighted in Fig. 6. The portfolio manager's role should consist primarily of being the caretaker of the M&S portfolio, which includes deciding when the portfolio should be used to address an identified need, understanding what the portfolio requires to address this need, and verifying and validating that the components in the portfolio meet the necessary performance and interoperability requirements. Note that this role does not preclude the portfolio manager from being an M&S co-developer or participating in defining the overall M&S strategy. Indeed, the portfolio manager should be a full participant on the leadership team of any system program office. The portfolio manager role also includes helping to design field tests and correctly integrating the output of those tests with the results of M&S to achieve the goals of the given work task.

Potential APL Role

The M&S portfolio manager will be central to the success of this process and should be an organization that has the breadth of capability to understand the system in all its complexity, from physics to mathematics to engineering, and also the ability to integrate system concepts into mission-level impacts. This organization should have strong software engineering capabilities and be effective at managing processes and coordinating participation from multiple organizations. We also believe that the M&S portfolio manager should be independent and able to provide guidance to the government free from conflicts of interest. Agile systems engineering may yield "rudder changes" throughout the course of system development that are in the best interest of the sponsor but may be undesirable for system developers for a myriad of reasons. For example, some program managers may not want realistic models because they may "make the program look worse. The performance of a weapon can deteriorate when factors such as turbulence, terrain, or targets are factored in. If the models are 'too realistic,' the estimates look worse."[15] This potential conflict of interest again argues for the involvement of an independent organization.

APL has a unique opportunity to take a stronger role in the systems engineering activities of our sponsors. Given our organizational strengths and values, the M&S portfolio manager role described herein is a good fit for us. As we look to the future for all of our (current and potential) sponsors, we should consider the potential benefits of the ideas presented herein and identify opportunities to apply them as appropriate.

SUMMARY

In recognition of recent DoD trends and feedback from sponsors, we suggest an application of agile practices to systems engineering processes through somewhat of a paradigm shift for the use of M&S in system development and deployment. There is great potential for an integrated M&S-based systems engineering process managed by an independent agent to save money, increase efficiency, and ensure that the right decisions are being made as new systems are developed and deployed. The fact that the approach is grounded in physics, and that fidelity can be customized to particular applications, helps to ensure that the predicted results are believable and can easily be modified to tackle a wide variety of systems engineering problems. In addition, ensuring that M&S components are interoperable will increase the power and flexibility of the M&S portfolio to answer more questions "out of the box" as system development matures. The reusability and scalability of the portfolio should give great insight to system designers, architects, and end users alike and, by its design, should be able to overcome the many roadblocks that SBA faced and leverage the lessons learned from past attempts at SBA implementation.

We realize that the ideas and concepts presented in this article cast a large net and, in some cases, are a bit utopian. Some of our claims (e.g., in the long run this will save money) are improvable *a priori* (although there are several smaller-scale, in-house use cases that give us confidence in the approach and a plethora of quantitative evidence from the software community that agile approaches work). The transformation described herein will require whole-scale commitment to realize optimal benefit, and indeed the creation and management of such a process is not trivial. It will require everything from a broad understanding of the use of systems at the mission level down to detailed technical expertise in the way that subsystems operate and interact. It will require world-class expertise in the design and building of M&S and an intimate knowledge of current standards of M&S interoperability. It will necessitate big-picture thinking, accurate anticipation of future trends, rigorous project management, and attention to detail. Indeed, these factors present a formidable challenge, but organizations like APL are uniquely positioned to bring the independence and technical credibility to make it work.

REFERENCES

[1]U.S. Government Accountability Office, *DOD Acquisition Outcomes: A Case for Change*, GAO-06-257T, http://www.gao.gov/new.items/d06257t.pdf (15 Nov 2005).

[2]Turner, R., "Toward Agile Systems Engineering Processes," *CrossTalk: The Journal of Defense Software Engineering*, http://www.stsc.hill.af.mil/CrossTalk/2007/04/0704Turner.html (Apr 2007).

[3]Hass, K. B., "The Blending of Traditional and Agile Project Management," *PM World Today* **IX**(V), 1–8, http://www.pmforum.org/library/tips/2007/PDFs/Hass-5-07.pdf (May 2007).

[4]Wikipedia contributors, "Systems Engineering," *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/Systems_engineering (accessed Apr 2009).

[5]Schlager, J., "Systems Engineering: Key to Modern Development," *IRE Trans*. **EM-3**(3), 64–66, doi: 10.1109/IRET-EM.1956.5007383 (Jul 1956).

[6]Cloutier, R., "Migrating from a Waterfall Systems Engineering Approach to an Object Oriented Approach—Lessons Learned," in *Proc. of International Council on Systems Engineering (INCOSE) Region II Conf*., Las Vegas, NV, http://www.calimar.com/Papers/RobINCOSE-ICSE2004Final2.pdf (15 Sept 2004).

[7]Defense Acquisition University website, http://www.dau.mil/default.aspx (accessed 25 Jan 2008).

[8]Department of Defense, *Guide for Integrating Systems Engineering into DoD Acquisition Contracts*, Version 1.1, http://www.acq.osd.mil/sse/docs/Integrating-SE-Acquisition-Contracts_guide_121106.pdf (11 Dec 2006).

[9]Beck, K., Beedle, M., van Bennekum, A., Cockburn, A., Cunningham, W., et al., *Manifesto for Agile Software Development*, http://www.agilemanifesto.org/ (2001).

[10]U.S. House of Representatives, "Recognizing the Contribution of Modeling and Simulation to the Security and Prosperity of the United States," *Congressional Record*, **153**(113), H7807–H7811, House Resolution 487, http://www.govtrack.us/congress/bill.xpd?bill=hr110-487 (16 Jul 2007).

[11]Keane, J. F., Lutz, R. R., Myers, S. E., and Coolahan, J. E., "An Architecture for Simulation Based Acquisition," *Johns Hopkins APL Tech. Dig*. **21**(3), 348–358 (2000).

[12]Davis, W. J., "Simulation-Based Acquisition: An Impetus for Change," in *Simulation Conf. Proc. 2000*, Orlando, FL, pp. 1061–1067 (2000).

13Zittel, R., "The Reality of Simulation-Based Acquisition—and an Example of U.S. Military Implementation," *Acquisition Review Quarterly*, **Summer 2001**: 121–132 (2001).

14Konwin, K. "Simulation Based Acquisition: The Future Way DoD Will Do Business," *National Summit on Defense Policy, Acquisition, Research, Test, and Evaluation Conf.*, http://www.dtic.mil/ndia/2001summit/konwin.pdf (26 Mar 2001).

15Erwin, S. I. "Simulation-Based Acquisition Only A 'Slogan', Says Survey," *National Defense Magazine*, http://www.nationaldefensemagazine.org/archive/2001/November/Pages/Simulation-Based6926.aspx (Nov 2001).

16Maurer, F., Martel, S., "On the Productivity of Agile Software Practices: An Industrial Case Study," *CiteSeer*, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.1925 (2002).

17Wikipedia contributors, "SOA Governance," *Wikipedia, The Free Encyclopedia*, http://en.wikipedia.org/wiki/SOA_Governance (accessed Mar 2009).

18Windley, P. J., "SOA Governance: Rules of the Game," *InfoWorld.com*, **Special Report**: 29–35, http://akamai.infoworld.com/pdf/special_report/2006/04SRsoagov.pdf (23 Jan 2006).

19U.S. Department of Transportation, *Systems Engineering for Intelligent Transportation Systems*, Publication No. FHWA-HOP-07-069, http://ops.fhwa.dot.gov/publications/seitsguide/seguide.pdf (Jan 2007).

The Authors

Timothy M. Frey is a member of APL's Senior Professional Staff and is Assistant Group Supervisor of the Special Concepts and Engineering Group of the Global Engagement Department. He holds a B.A. in mathematics and music from the University of Richmond, a master's of statistics from North Carolina State University, and an M.B.A. from The Johns Hopkins University Carey Business School. At APL, he has spent considerable time in weapons systems analysis and command and control, and he currently helps to lead APL's Interactive Simulations Program. Mr. Frey's main areas of interest include M&S-based systems engineering, agile methods, end-to-end concept development, mentoring, business development, and strategic planning. **Matthew C. Valencia** is a member of the Senior Professional Staff and a Section Supervisor in the Special Concepts and Engineering Group of the Global Engagement Department. He holds a B.S. in physics from Carnegie Mellon University and an M.S. in computer science from The Johns Hopkins University. He is the Project Manager for the SEAL Delivery Vehicle Desktop Trainer, and his areas of interest include M&S, software development, and agile methods. For further information on the work reported here, contact Timothy Frey. His e-mail address is timothy.frey@jhuapl.edu.

Timothy M. Frey

Matthew C. Valencia

The *Johns Hopkins APL Technical Digest* can be accessed electronically at **www.jhuapl.edu/techdigest**.