SYM-AM-18-066

PROCEEDINGS
OF THE
FIFTEENTH ANNUAL ACQUISITION RESEARCH SYMPOSIUM

WEDNESDAY SESSIONS
VOLUME I

Acquisition Research:
Creating Synergy for Informed Change

May 9–10, 2018

Published April 30, 2018

Approved for public release; distribution is unlimited.

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL

Enabling Operationally Adaptive Forces

Dan Boger—served in the U.S. Navy from 1968 to 1975. He earned his PhD at the University of California, Berkeley in 1979, when he joined the faculty at the Naval Postgraduate School. He currently serves as the Chairman of the Information Sciences Department. [dboger@nps.edu]

Charles Deleot—served in the U.S. Navy from 1967 to 1990. He holds three master's degrees in management and IT related fields. He served as a senior civilian advisor on the staff of the Commander, Pacific Fleet from 1970 to 2002. He serves as the President of the Patriot Foundation in Southern Pines, NC. This nonprofit has raised over $5 million for dependents of killed or wounded Army Special Force soldiers. [cdeleot@nc.rr.com]

Norman Eaglestone—is a senior software engineer who has supported Open Management Group standards development since the 1990s. He served at various IT related companies, including Sun Microsystems, Northup Grumman, SAIC, and Sierra Nevada. [eaglestoneconsultingllc@gmail.com]

Scot Miller—is a Faculty Associate–Research at the Naval Postgraduate School. He served in the U.S. Navy from 1978 to 2007 and at the Naval Postgraduate School since. He holds master's degrees in three separate fields. [samille1@nps.edu]

Scott Rosa—is currently a Major serving in the U.S. Marine Corps. He is a graduate student at the Naval Postgraduate School. [sjrosa@nps.edu]

Abstract

Emerging warfare trends demand an operationally adaptive force, ready to adjust material solutions, such as systems and systems integrations, in near real time. Software is the most important element of those changes. The DoD has a poor track record with software development, as well as in requirement development, semantic interoperability, and cyber awareness and defense, to name just a few. Mechanical and aeronautical engineering migrated to machine-based designing and testing two decades ago, with transformative results. Software engineering has lagged in this transformation, but our research shows that it has reached the tipping point. What does this mean for the DoD? It means that formal models will enable very rapid capability development, integration, test and evaluation, semantic interoperability, and cyber assessment and remediation, changing the way the entire DoD acquisition enterprise performs. We envision a virtuous improvement cycle where costs spiral down, speed to capability accelerates, and performance increases, all due to formal models.

Introduction

General McChrystal authored a book in 2016 about fighting the global war on terror. He concluded his best successes occurred when he and his forces rapidly adapted, since his lesser experienced, lesser resourced foes always changed their tactical approach (Collins et al., 2015).

These foes enjoyed a faster adaptiveness loop at first than McChrystal's forces, but as his experience grew, his forces, too, learned to be adaptive—adaptive not only in adjusting tactics, but in using equipment, systems, and applications differently (Collins et al., 2015). This often was a struggle, as the cumbersome acquisition and support processes struggled to keep up. Engaging emerging near-peer threats in the future will demand an even more resilient and adaptive U.S. military force.

Our research team postulates that if reforming this lumbering set of processes (requirements, procurement, test, cyber, etc.) was possible, it would enable all operators to be adaptive in near real time. Serving as acquisition professionals, IT engineers, and operators at the tactical edge, this research team has over 200 years of related experience. Our first realization was that nearly every technical requirement needing adaptiveness relied in whole or in part on software, so that is where we focused our efforts.

We also learned, and proved, that yes, it is quite possible to grow an adaptive acquisition enterprise. What follows is why, how, and what remains to be done.

Operational Adaptability

We start with *what is operational adaptability*? The research team established four broad requirements.

First, achieve rapid requirements collection. Operators often know soonest when something needs help, change, or adjustment. We still need to capture that emerging requirement and insert it into the acquisition-related processes, including validation and funding.

Second, given a validated and funded requirement, accelerate problem fixing or new capability delivery. Today that means months or years; we need to be like Google, incorporating a nearly continuous cycle of improvement! That implies an equally responsive test and evaluation approach as well.

Third, leverage current legacy applications and data sources. This means rapid integration as well as platform provisioning. The DoD invests billions in systems, so reusing them makes sense as they do have value.

Finally, account for cyber impacts. Introducing new capabilities and novel integrated systems-of-systems mash-ups means delivering potentially vulnerable systems, where the operational risk is not understood. That would be unsatisfactory. Any reform to our acquisition processes means enabling rapid risk management and corrections.

These four components need to work together. When an actual requirement is identified, the DoD needs rapid validation and funding to support the agents responsible for an attentive response. Careful consideration of legacy apps and data needs to be included, while none of this should proceed without including cyber defense and awareness as part of the overall process. Our DoD processes need to be highly integrated and supportive of one another.

Acquisition and Related Processes as an Adaptiveness Enabler Today

So how are our monolithic acquisition processes doing now, compared to these four components? In a word, terrible. We won’t repeat the disaster stories of many programs, but there are many. How do our current approaches match up to the vision outlined above?

Most agree that requirements are tricky. Operator input is a must, yet often operators just want to slightly improve their current capabilities, and are completely unaware of technical opportunities. That makes sense, of course, since they do have real work to do. Incorporating emerging capabilities is a must as well, though. Too often it seems that the DoD wants to adopt the newest IT technologies without enough thought. For instance, in 2003 FORCEnet was the big C4I theme of the day in the Navy. Gray beard technologists recommended a service-oriented architecture. The first page of every SOA book says, “Naturally, don’t try SOA if you are not working in a business with a well-connected network and well understood business processes” (Brauel et al., 2009). The Navy discovered that intermittent satellite links do not equate to a well-connected network. Our doctrine was well understood, but seldom followed! This was not a recipe for SOA implementation. The research team believes the same misunderstandings exist today in the DoD for jumping on the Cloud and AI “bandwagons.” We are on the precipice of making grave mistakes in our IT investments. We certainly do not have the rapid requirements capture and funding process needed.

For rapid capability delivery, there are actually many examples of success. However, just about all of these required high-level phone calls, tons of money, and the transport of expensive technical tiger teams to faraway places. That approach is unsustainable. For most programs, the answer is no, it takes much longer than it should, by anyone’s measure. Consider friends of the researchers in PEO C4I’s PMW-150. In 2008 they rapidly built the Command and Control Rapid Prototyping Continuum, with software engineers embedded with operators. The operational level customer was thrilled with the rapid delivery and impressive results (Fein, 2011). Yet, has that translated to an afloat capability? Almost. But it is a decade later despite the fact that this organization is filled with consummate professionals, forward leaning technicians, and outstanding leadership.

Integrating capabilities today is quite the challenge. Two factors play here. First, if one just integrated one additional system, it is straightforward. Integrating to two is a bit harder. Integrating to five though, proves an N-squared relationship between the number of systems/data sources to be integrated and the number of connections needed. Add maintaining configuration management of all this, and the challenge grows geometrically in yet another dimension.

For cyber, can we agree that our older approaches leave much to be desired? The Risk Management Framework (RMF) process today adds emphasis on early cyber engineering and requires continuous monitoring, steps in the right direction (DoD Instruction, 2014). Yet the process still is time consuming.

Today, each of these components is an independent silo. Yes, RMF is designed to be included in the capability delivery silo, and this is slowly occurring. The remaining silos barely touch, yet are completely dependent on each other. Each silo has a different boss. For instance in the Army, TRADOC owns requirements. Capability delivery is the province of service acquisition and program executive officers. Integration is also their province. Cyber approval for RMF is led by service cyber commands, completely independent of the service acquisition executive and PEOs. Making this responsive is difficult even in the best of times.

Foundational Approach

We researched if our imagined operationally adaptive acquisition process was technically feasible. Our findings were successful. Figure 1 shows the traditional approach at the top and the revised approach at the bottom. This is an example drawn from the mechanical engineering community. Traditionally (at the top of the figure), engineers designed parts and drafted formal drawings for the machinist who converted the drawings into actual prototypes. Next, the part was iteratively tested and improved, until a set of standards were met. Often mechanical engineering students were required to intern on the machine shop floor so they could appreciate the difficulties of translating their drawing into actual parts.

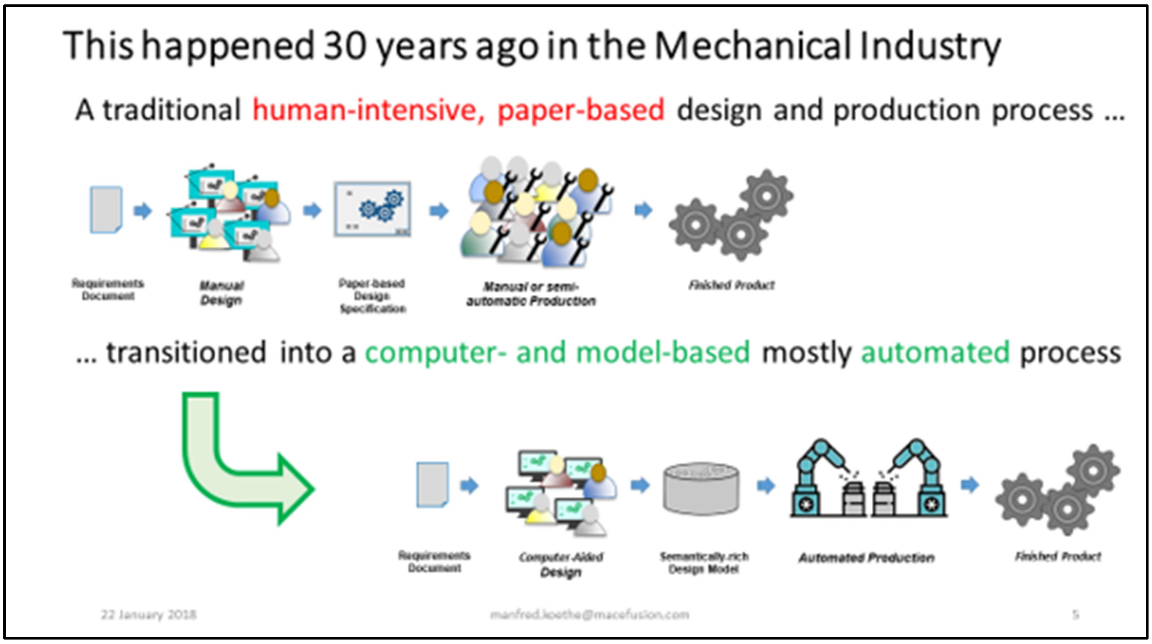

Figure 1. Machine-Based Engineering Transformation
(Koethe, 2017)

The bottom of Figure 1 shows how parts are produced today. Yes, the mechanical engineer produces drawings, but these leverage computer aided design/computer aided manufacturing (CAD/CAM) practices. What comes to the machine shop floor is a digital product. This product undergoes extensive testing in a virtual environment, so there is high confidence the part will work. The machinist programs a robot machinist that produces the part to exacting standards.

We learned that as late as the 1990s, the transmission in most cars was unique, because of the variability in the precision of the manufacture of that transmission. Fixing a transmission meant identifying that a part in the transmission was bad, then replacing the whole transmission. Since tolerances were so tight, parts were not interchangeable (M. Koethe, personal communication, 2017). That is not the case today, no matter what AAMCO says.

Read *Aviation Week* and you will realize that the aeronautical engineering community does the same (Bozdoc, 2006). Even the prototype for our most modern fighter, the F-35, flew within three years of contract award. It took an additional 14 years to create the operational software ("F-35 Initial," 2013)!

We asked ourselves, where is the software engineering equivalent to this mechanical/aeronautical engineering approach? We found it under our noses. The answer is formal software modeling, which is the software development equivalent to engineering CAD/CAM development.

In this approach, software engineers, coders, etc., create models of the functionality they want to develop. Once this formal representation is achieved, it is transformed, depending on the hardware selected, into a true formal model. Automatic code generators produce code, then assess code quality. Once satisfied, this code is provisioned onto the designated platform.

This formal model, just like the digital representation of a mechanical part, can be tested in a virtual environment, including a cyber-environment. "What-if" and engineering trade-off analyses can be easily achieved using parametric modeling. A library of endpoints, which are agreements between this kind of formal model and existing application protocols and data sources, makes integration much faster (N. Eaglestone, personal communication, 2014).

Automatic code analysis tools ensure the code produced is optimized for quality, which is directly related to security. New tools in the development environment enable each integration of systems of systems to achieve semantic interoperability, which is central to achieving success in machine learning, deep learning, and other artificial intelligence techniques.

This approach allows humans to do what they do best: consider all options and employment considerations, understand the operating environment, and address constraints. It allows machines to do what they do best, which is to keep this information for future use and reuse, and to produce code at least 100,000 times faster than humans (Eaglestone, 2012).

What is most promising about these tools and their power is that it will easily enable collaboration between each of the tools' users. Integrating these tools means an enterprise approach to solving the operational adaptation challenges.

While not complete, this approach also enables an emerging capability that is more icing on the cake. Additions to systems engineering processes include designing for man-machine interdependence. Achieving such interdependence, through careful consideration of how the observability, predictability, and directability between men and machines can be achieved, is difficult. This new systems engineering addition allows for establishing the requirements to achieve this interdependence. Such an approach can be easily incorporated into the tools described above (Johnson, 2014). An ability to achieve interdependence (another word is *collaboration*) between man and machine might even support a fourth offset strategy.

Additional Details

Our research uncovered the technical tools to produce a revolution in military affairs. Imagine actually being able to respond in hours or at most, days, to pressing operational needs? This would be a game changer.

These tools are based on open standards developed by the Object Management Group (OMG). Many of your contractor companies send representatives to their meetings (R. Soley, personal communication, 2015). This is not magic, but rather a set of tools that have evolved over the past 20 years and have reached maturity. They are ready to be employed today!

These tools now enable semantic interoperability between integrated systems. This is a huge accomplishment, yet few programs are leveraging this capability. Previously, semantic interoperability could be achieved through very expensive and time consuming one off programming and was brittle to configuration changes. OMG adopted a new Archetype Modeling Language, born from efforts to integrate various health care systems, to achieve semantic interoperability. Our research shows that creating meaning between medical systems is at least as hard as doing so for DoD systems (N. Eaglestone, personal communication, 2016).

We reviewed six separate efforts that used formal modeling approaches. Their project requirements varied from building a simple set of models evaluating counter-battery fire, developing a web portal that assesses software for quality and cyber resilience, and translating Chinese notice to mariner's messages for U.S. nautical chart changes. Table 1 summarizes the type of capabilities and integration required, how long it took to produce, and estimated costs.

As this table shows, these are remarkable project achievements executed in very short amounts of time. Keep in mind that these projects required expert formal modelers. Much of their time was spent building the model; they draft very little actual code. What code they do write is often associated with transforms in the modeling process, not actual functionality of the systems.

Four of the six were purely proofs of concepts, where in every case the sponsor was very satisfied with the results. The cartography project has continued at the National Geospatial Intelligence Agency (NGA), since it was the only viable solution to a growing cartography correction crisis (R. Wicks, personal communication, June 12, 2015). This code assessment portal is under refinement, including a component that would enable continuous code monitoring, an RMF requirement.

One final exciting piece of formal modeling is the ease in producing transforms that convert the formal model into any required program documentation (N. Eaglestone, personal communication, October 12, 2017). For instance, most programs are required to deliver various DoDAF views. That is a simple, minute-long process using formal models.

Table 1. Formal Model Proofs of Concept, 2012–2018

(N. Eaglestone, personal conversation, 2017)

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

Challenges

New processes are not without challenges. We uncovered five significant issues to start with; no doubt there may be others.

First, as one might imagine, creating a formal model is not easy. It takes many iterations between operators and modelers to get the model right. Many program managers grow very impatient with the rate of progress. Therefore, many are unwilling to risk trying an approach that promises such great deliverables but has “nothing” (since there is only a model) to show for months. Our research shows that patience does pay off. Proper prior preparation prevents poor performance. That has been an axiom for software development since the 1950s. Formal modeling is just that to an extreme. But how can that be proven to program managers?

Observers of model-based systems engineering will point to large DoD efforts that focused on using formal models for their objectives, but without any success. For instance, in the early 2000s, the Navy led the Single Integrated Air Picture initiative, which was model-based. No doubt quality engineers and modelers combined with operational experts to create the models. However, the modeling expertise, the standards, and the tools were not quite mature enough to guarantee success (Dinkins, 2006). We believe that the standards and the tools are more rigorous now, and creating the formal models needed to achieve rapid capability delivery are now present.

This leads directly to the third problem. Even today, creating formal models requires the work of master modelers, not journeymen engineers fresh from engineering school. OMG is working on methods to improve the tools that would help solve this conundrum (M. Koethe, personal communication, December 14, 2017). Right now, expert formal modelers demand high hourly rates and receive them. The DoD's own contracting guidelines often prevent us from being able to hire these masters. It's a chicken and egg challenge. How can we train apprentice engineers to be masters if we cannot hire the masters to train them?

The fourth issue relates back to empowering the operators as requirements creators. In a perfect world, operators, working directly with modeling masters, would give input. The modelers would then tease out exactly the meaning, then iterate again (and again) with the operators to ensure correctness. What tools exist to support this process? So far, the answer is very few. In the cartography proof of concept a PowerPoint-based tool, generated by the model itself, was used to provide the operators an idea of what was going on, enabling them to provide feedback (Wicks, 2015). More intuitive tools are needed. Of course, direct interaction between modelers and operators cuts out the entire requirements validation and funding process.

This points toward the last big challenge: Who is in charge of all of this? The service acquisition executive? The individual program managers? The Program Executive Officers? The service's Pentagon staffs? Our research uncovered several possible options, but the common thread was that someone must lead the effort for a long period of time, at least 10 years. This implies someone of passion who is unusually adept in DC political wrangling. The Army is discussing a new command that might actually try to do this (L. Brown, personal communication, January 17, 2018). It bears watching over this summer to see what they decide.

Conclusions and Next Steps

It is not just our research team that explores formal models. An Army team investigated helicopter flight control software by releasing a formal model in their request for proposals, as a sort of acquisition experiment. They used a cost-estimating team to predict what the bids would be, both in time and money. Four proposals said they would do the work for one-fourth the anticipated amount, in half the time. The Army funded three and they all beat their predictions with working software (A. Farrar, personal communication, May 17, 2017).

Formal modeling is real. It works, and it will reduce costs, accelerate delivery, and improve operational performance. Achieving formal models enables parametric modeling, rapid test and evaluation, semantic interoperability, and improved code, all while enabling operators more intimate requirements inputs. This creates a virtuous cycle of continuous improvement in all phases of the requirements, procurement, test, sustainment, and cyber defense processes, with automated document generation.

We invite the readers to join us in our quest to make acquisition the chief enabler of operationally adaptive forces. Please send us any other ideas on how this can occur.

References

Bozdoc, M. (2006). History of computer aided design. Retrieved from http://www.thocp.net/software/software_reference/cad_bozdoc.htm

Brauel, B., Matsumura, M., & Shah, J. (2009). *SOA adoption for dummies*. Hoboken, NJ: Wiley.

Collins, T., Fussel, C., & McChrystal, S. (2015). *Team of teams: New rules of engagement for a complex world*. London, UK: Portfolio.

Dinkins, R. (2006). *Single Integrated Air Picture (SIAP)* [PowerPoint slides]. Retrieved from https://www.ll.mit.edu/HPEC/agendas/proc06/Day2/03_Fairbairn_Pres.pdf

DoD. (2014). *Risk Management Framework (RMF) for DoD information technology (IT)* (DoD Instruction 8510.01) Retrieved from http://www.esd.whs.mil/Portals/54/Documents/DD/issuances/dodi/851001_2014.pdf

Eaglestone, N. (2012). *Universal command and control* [PowerPoint slides].

F-35 initial operational capability [Report to Congressional Defense Committees]. (2013). Retrieved from https://breakingdefense.com/documents/services-set-ioc-dates-for-f-35s-confidence-is-the-watchword/

Fein, G. (2011). *ONR develops new acquisition model for delivering information to the fleet.* Retrieved from https://www.onr.navy.mil/en/Media-Center/Press-Releases/2011/C2RPC-Information-Fleet

Johnson, M. (2016). *Co Active Design* [PowerPoint slides]. Retrieved from http://www.ihmc.us/users/mjohnson/publications.html

Koethe, M. (2017). *Model oriented development environment* [Power Point slides].

Wicks, R. (2015). *Notice to Mariners (NtM) Production Automation Limited Objective Experiment (LOE)* [PowerPoint slides].

ACQUISITION RESEARCH PROGRAM
GRADUATE SCHOOL OF BUSINESS & PUBLIC POLICY
NAVAL POSTGRADUATE SCHOOL
555 DYER ROAD, INGERSOLL HALL
MONTEREY, CA 93943

www.acquisitionresearch.net