Achieving Better Buying Power through Cost-Sensitive Acquisition of Open Architecture Software Systems

Walt Scacchi is senior research scientist and research faculty member at the Institute for Software Research, University of California, Irvine. He received a Ph.D. in Information and Computer Science from UC Irvine in 1981. From 1981-1998, he was on the faculty at the University of Southern California. In 1999, he joined the Institute for Software Research at UC Irvine. He has published more than 150 research papers, and has directed more than 65 externally funded research projects. In 2011, he served as Co-Chair for the 33rd Intern. Conf. on Software Engineering—Practice Track, and in 2012, he served as General Co-Chair of the 8th IFIP International Conference on Open Source Systems.

Thomas Alspaugh is a project scientist at the Institute for Software Research, University of California, Irvine. His research interests are in software engineering, requirements, and licensing. Before completing his Ph.D., he worked as a software developer, team lead, and manager in industry, and as a computer scientist at the Naval Research Laboratory on the Software Cost Reduction or A-7 project.

Abstract

Our presentation focuses on our ongoing investigation and refinement of techniques for identifying and reducing the costs, streamlining the process, and improving the readiness of future workforce for the acquisition of complex software systems. Emphasis is directed at identifying, tracking, and analyzing software component costs and cost reduction opportunities within acquisition life cycle of open architecture (OA) systems, where such systems combine best-of-breed software components and software products lines (SPLs) that are subject to different intellectual property (IP) license requirements.

The Department of Defense, other government agencies, and most large-scale business enterprises continually seek new ways to improve the functional capabilities of their software-intensive systems. The acquisition of OA systems that can adapt and evolve through replacement of functionally similar software components is an innovation that can lead to lower cost systems with more powerful functional capabilities. Our research identifies and analyzes how software component costs and IP license requirements interact to drive down (or drive up) total system costs across the system acquisition life cycle. The availability of such new scientific knowledge and technological practices can give rise to more effective expenditures of public funds and improve the effectiveness of future software-intensive systems used in government and industry. Thus, a goal of this presentation is to support and advance a public purpose through acquisition research and results.

Overview

The Department of Defense, other government agencies, and most large-scale business enterprises continually seek new ways to improve the functional capabilities of their software-intensive systems with lower acquisition costs. The acquisition of open architecture (OA) systems that can adapt and evolve through replacement of functionally similar software components is an innovation that can lead to lower cost systems with more powerful functional capabilities. OA system acquisition, development and deployment are thus seen as an approach to realizing Better Buying Power (BPP) goals for lowering system costs while improving competition.

Our research identifies and analyzes how new software component technologies like apps and widgets for Web-based and/or mobile devices, along with their intellectual property (IP) license and cybersecurity requirements interact to drive down (or drive up) total system costs across the system acquisition life cycle. The availability of such new scientific knowledge and technological practices can give rise to more effective expenditures of public funds and improve the effectiveness of future software-intensive systems used in government and industry. Thus, a goal of this presentation is to explore new ways and means for achieving cost-sensitive acquisition of OA software systems, as well as identifying factors that can further decrease or increase the costs of such systems at this time.

We begin by briefly reviewing to identify a set of recent trends in the development of OA software systems that intend to develop more capable OA systems. These trends include the transition to adoption of small-form factor software components as distinct applications ("apps") and widgets that exploit modern Web capabilities. We then turn to examine some key goals of the BBP 2.0 initiative that direct attention to adoption of OA system development practices that affect acquisition practices. Next, we identify a new set of emerging challenges to achieving BBP through OA software systems. We then identify three new practices to realize the cost-effective acquisition of OA Software systems.

Recent Trends Affecting Better Buying Power through OA Systems

We find there are four broad trends that mediate the cost-effectiveness and buying power of emerging OA system acquisition efforts. These include: (a) the move towards shared, multi-party acquisition and agile development of new OA systems across compatible software ecosystems; (b) exploitation of new software component technologies compatible with Web and mobile devices; (c) growing diversity of cybersecurity challenges to address during system development; (d) new software development business models for app/widget development and deployment. Each is examined in turn.

A. Multi-party acquisition and development system ecosystems –
Many in the Defense community seek to embrace the acquisition and development of agile command and control (C2) and related enterprise systems

[GBC14, GMH13, GuW12, RBS12, ScA13, ScB12]. Such systems are envisioned to arise from the assembly and integration of system elements (application components, widgets, content servers, networking elements, etc.) within a software ecosystem of multiple producers, integrators, and consumers who may supply or share the results of their efforts. The assembly and integration of system elements produces “C2 system capabilities” (C2SCs). Our purpose is to identify how our approach to the design of secure OA systems can be aligned with this emerging vision for agile C2 system development and adaptive deployment. Along the way we focus on design of OA system capability involving office productivity components that must be configured as a secure C2SC.

The design and development of agile C2 systems follows from two sets of principals: one set addressing guidelines/tenets for multi-party engineering (MPE) of C2 system components; the other set addressing attributes of agile and adaptive ecosystems (AAE) for producing C2SCs or C2 system elements. For brevity, we identify the principals for MPE and AAE, as they are more fully explained elsewhere [RBC12], but we do so in ways that foreshadow and more clearly align with our approach that follows in later sections.

MPE Tenets:

- Provide small system components that can be rapidly developed, and accommodate different functionally equivalent variants, or functionally similar versions (software product lines).
- Certify components are consistent with “shared agreements” regarding security requirements, system architecture, data semantics, production and integration processes or process constraints, and other aspects of mission-specific or mission-common domain models.
- Supply diverse C2 system components via a market of component producers or system integrators.
- Assemble and integrate C2SCs from components available in the market that are consistent with relevant shared agreements.
- Provide feedback from C2 system users to component producers or capability integrators to improve market efficiency and effectiveness.

AAE Attributes:

- Encourage and sustain a software ecosystem that is agile (supports assembly and integration C2SC) from components in market, and adaptive (supports substitution of functionally similar component versions or functionally equivalent component variants), in line with user feedback.
- Component markets are federated so as to accommodate sharing, reuse, or trading of components across system integrators or user organizations.
- Shared agreements serve as a basis for enabling multi-party collaboration in system development, integration, and evolution/sustainability.
- Production, integration, or post-deployment support for components or C2SCs must be viable for small businesses or large, as well as promoting

market diversity and effectiveness.
- Consumer/user organizations seek to manage portfolios of components or C2SCs that collectively improve mission effectiveness, agility and adaptiveness, while reducing costs.

Subsequently, to help understand what we mean by a software ecosystem, we use Figure 1 to represent where different parties are located across a generic software ecosystem, and the supply networks or multi-party relationships that emerge to enable the software producers to develop and release products that are assembled and integrated by system integrators for delivery to end-user organizations.

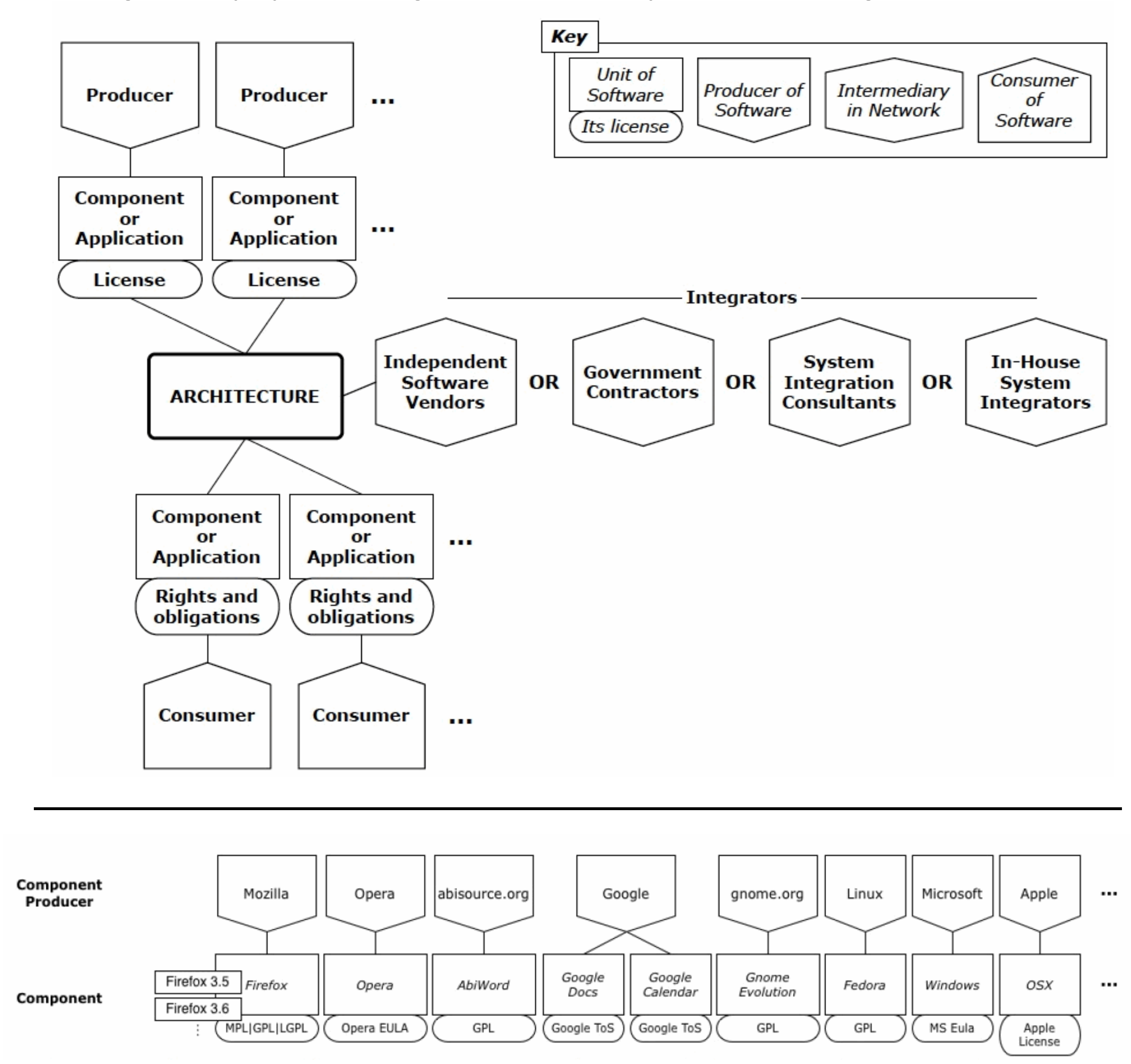


Figure 1. A generic software ecosystem supply network (upper part), along with a sample elaboration of producers, software component applications, and licenses for an OA system components they employ (lower part) [ScA12a].

As noted, OA system components can include software applications (apps) and widgets. Widgets are lightweight, single-purpose web-enabled applications that users can configure to their specific needs [Giz11, GMH13, ScA13b]. Widgets can provide summary information or a limited view into a larger application that can be used alongside related widgets provides an integrated view, as required by users.

The lower part of Figure 1 also identifies where elements of shared agreements like IP licenses or cybersecurity requirements enter into the ecosystem, and how the assembly of components into a configured system or subsystem architecture by system integrators effectively (and perhaps unintentionally) determines which IP license or cybersecurity obligations and rights get propagated to consumer or end-user organizations. Agreement terms and conditions acceptable to consumer/end-user organizations flow back to the integrators. This helps reveal where and how shared agreements will mix, match, mashup, or encounter semantic mis-matches at the system architecture level, which is one reason why we use (and advocate) explicit OA system models.

Overall, a move towards MPE and AAE substantiates a path towards decentralized OA system development, integration, and deployment [DoD12, Giz11, SBH12]. This decentralization will in turn engender acquisition and development of heterogeneously-licensed systems (HLS), whereby different software components (apps, widgets) will be subject to different IP licenses [AIA13, AIS10], as well as to different cybersecurity requirements [DAU-CVE14, ScA12b, ScA13a, ScA13b, ScA13c]. This in turn implies that such components, their IP licenses, and cybersecurity requirements will be subject to ongoing evolution across a diversity of methods, shown in Figure 2 [ScA12a, ScA13b].

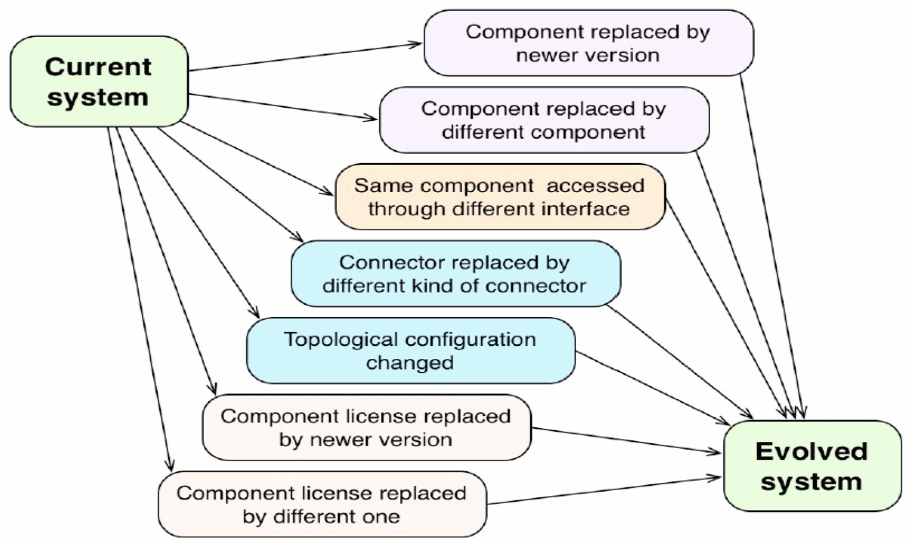

Current system
Component replaced by newer version
Component replaced by different component
Same component accessed through different interface
Connector replaced by different kind of connector
Topological configuration changed
Component license replaced by newer version
Component license replaced by different one
Evolved system

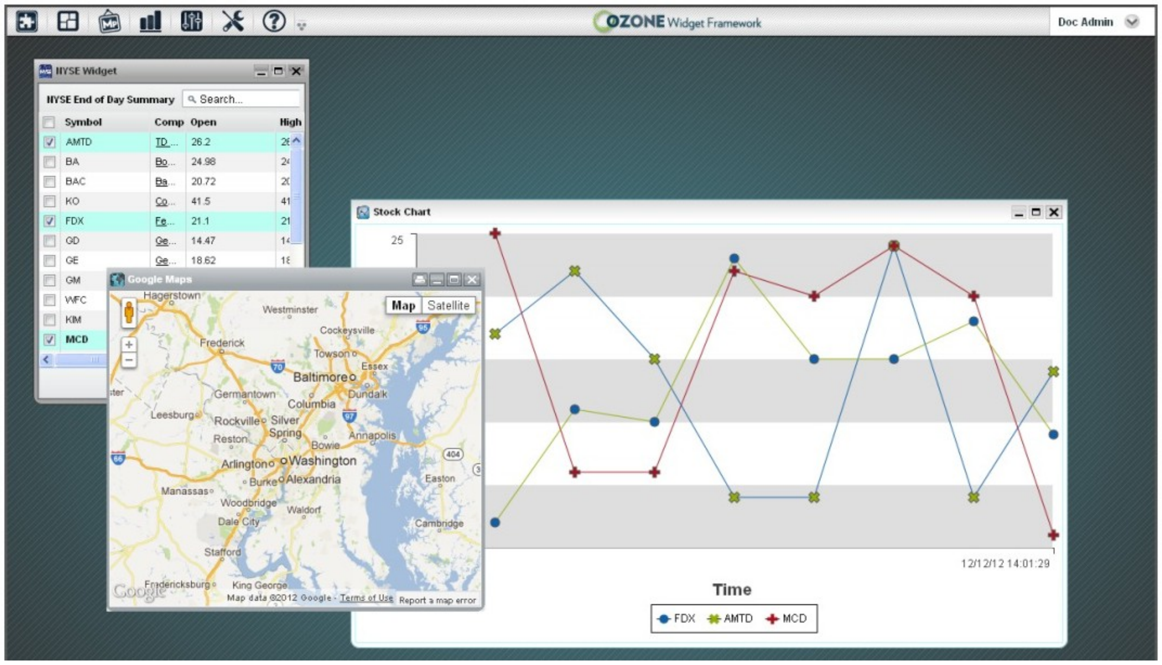
Figure 2. Different kinds of common evolutionary changes that arise during OA software component development, deployment and sustained usage.

Heterogeneity, decentralization, cybersecurity and evolutionary dynamics will come to interact during OA system acquisition, development, and deployment.

These in turn will create a new generation of challenges for the acquisition workforce, in terms of training, new work and contract management practices, and need for automated assistance to track and manage oversight of policy compliance (e.g., for alignment with BPP and cybersecurity assessment). Without automated assistance, it appears that the acquisition workforce will be overwhelmed with technical details that interact with acquisition, development, and/or system integration contracts and software component IP licenses and cybersecurity requirements. Otherwise, these conditions suggest that acquisition management practices can complicate acquisition [GBC14], and thus potentially mitigate the benefits of BBP that can arise from MPE and AAE for C2 systems.

B. Moving towards shared development of Apps and Widgets as OA system components – Future OA systems for agile C2 may configured by system integrators, end-user organizations, or warfighters in the field. This would be accomplished through access to online repositories of software apps or user-interface widgets. The Ozone Widget Framework (OWF), a government open source software (GOSS) effort that is central to such agile OA system development. The OZONE family of products, includes the OWF and the OZONE Marketplace, the marketplace being an online repository whose operation is similar in kind to the online app stores by Apple and Google [ScA13b]. These products are built to fit the needs of human centered fusion activities in network centric warfare environments. The OZONE family of products is designed as a presentation layer toolkit that can be rapidly deployed in a variety of mission contexts ranging from strategic planning to enabling the creation of a real-time common operational picture and situation awareness applications. Figure 3 displays examples of OWF-based widgets operating in a Web browser, while Figure 4 shows OWF widgets deployed for use on a mobile device.

C. Growing diversity of challenges in cybersecurity – New types of software components like apps and widgets must be developed, deployed, and sustained in ways compatible with existing cybersecurity requirements. They must also be later adapted to accommodate emerging cybersecurity requirements that are not yet apparent. For example, there is growing interest in accommodating not just mobility, but also “Bring Your Own Device” (BYOD) capabilities. BYOD suggests that end-users and warfighters are bringing their own mobile devices with themselves into the field to support their mission. However, BYOD clearly exacerbates the technical challenges of cybersecurity assurance, often in ways that cannot be readily anticipated, as when independently developed component co-evolve in conflict to one another [Wei14]. Nonetheless, acquisition policy necessitates cybersecurity vulnerability and exposures be addressed [DAG14]. But at present, it is unclear what new kinds of requirements these new OA system components bring to the acquisition workforce. For example, a move to adopt mobile apps and/or mobile widgets means these OA system components must pass though an application security process for “vetting” these components.



NYSE End of Day Summary

	Symbol	Comp	Open	High
✓	AMTD	TD...	26.2	26
	BA	Bo...	24.98	24
	BAC	Ba...	20.72	20
	KO	Co...	41.5	41
✓	FDX	Fe...	21.1	21
	GD	Ge...	14.47	14
	GE	Ge...	18.62	18
	GM			
	WFC			
	KIM			
✓	MCD			

Figure 3. OWF Widgets running within a Web browser.

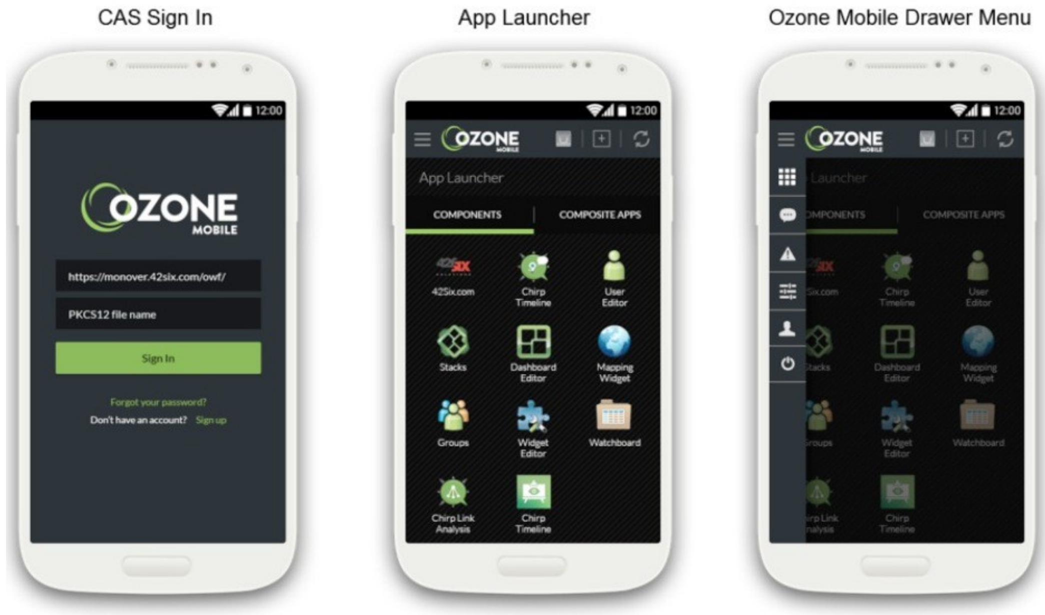


Figure 4. OWF Widgets running on a mobile device*.*

Vetting entails establishing what cybersecurity requirements are to be verified, how they are to be validated, as well as where, when and by whom these activities should be performed. One approach is to assume such vetting can be performed by a centralized authority, such as by the operator of the Ozone Marketplace. But it is not clear there will ever only be one such authority. Instead, if we foresee multiple marketplaces, which are already appearing both in GOSS and industrial online settings, then the acquisition workforce will be challenged in how best to determine which cybersecurity requirements must be addressed, validated, and compliance certified, as well as by whom and how often.

A move to widgets also presents new kinds of cybersecurity challenges when two or more widgets are configured together with one or more apps to create a mashup that provides an agile system capability. This situation refers to the technical challenges of inter-widget communication. Such component-component communication can be technically realized in different ways, such as via ad hoc, “open standards,” or publish-subscribe messaging interfaces, as well as whether point-to-point or as configured through a dynamic processing mashup [CFG13, End13]. While OA system guidance from the BBP 2.0 may stipulate reliance on “open standards” style widget interfaces and communications patterns be used, widget communication/interface standards/interfaces are still very new technologies and techniques. Thus, it is unclear which will survive and be widely adopted [End13a]. Similarly, knowledge about their proper usage is unclear, and thus is not yet ready for compliance assessment within current acquisition practices. The technical challenge is further complicated when apps/widgets are acquired from different online marketplaces. Different marketplaces may rely on different schemes for specification and interchange of shared data semantics between autonomously developed components. This in turn hinges on the expertise of OA system integrators, end-users, or warfighters to recognize how, where, and when the semantics of technical data interchange arise and to what consequences via component-component API alignments (to avoid mis-matches), data type representations, data formats (e.g., “CSV” vs. .xls vs. XML), data naming conventions (for resource discovery vs data modeling ontology), data range value limits, exceptional values, data-flow control signals, etc. These are still new technical problems that are yet to be readily resolved or to have development/usage guides.

D. New business models for OA software component development and use – New business models imply differentiated IP licenses and contracting practices. Given our discussion up to this point, along with reference to our recent acquisition research studies [AlA13, ScA11, ScA12b, ScA13b], this means different obligations and rights will be transferred from component producers to system integrators and end-user organizations. Some licenses are “buy and pay now,” while others are “free now, pay later, based on usage,” others are “many organizations (e.g., PEOs) will share purchase costs,” and so forth.

Acquisitions of new kinds of OA system components allow for new business models. These include new models for software component producers, system integrators, and end-user organizations. For example, new software and OA system development business models for software app/widget development and deployment include (in no particular order): franchising; enterprise licensing; metered usage; advertising supported; subscription; free component, paid service/support fees; federation reciprocity for shared development; collaborative buying; donation;

sponsorship; free/open source software (e.g., Government OSS – GOSS); and others [Hanf13]. Further, this list is not exhaustive; instead, it is only representative.

In contrast, for end-user organizations that involved in agile development of OA system components, or an integrated system capability, there is a need to developed and codify their own business models regarding OA software component development or system integration. These business models are constituted through "shared agreements" that allow for sharing the cost of component or integrated capability development and cybersecurity assurance vetting across multiple parties (e.g., multiple Program Offices). However, these shared agreements are also a core part of emerging MPE/AAE development practices. These agreements must convey how OA component development or system integration costs and security assurance will be shared, as well as how they will be sustained in the presence of interacting software component development, deployment, and evolution processes and practices [ScA13a]. Shared agreements denote the obligations the participating organizations are willing to accept, in order to realize the provided rights they need. So shared agreements can be expressed and assessed in the same manner, and with the same analysis tools and techniques, as IP licenses and cybersecurity requirements [ScA13b, ScA13c].

Software acquisition costs easily become difficult to predict/manage given diversity of business models, IP licenses, and implied software component cybersecurity assessment. Development/usage cost sharing agreements can further complicate determination of development cost, costs shares across organizations, and system costs over time as business models, component licenses, and cybersecurity assessment requirements evolve [ScA12a, ScA13a].

What kind of expertise do we expect the acquisition workforce to need in order to make adoption of "component-based system capabilities" (including for mobile devices) agile, adaptive, and practical across different commercial/governmental software marketplaces/ecosystems? What kinds of acquisition guidance is needed for articulating and streamlining Shared Agreements between multiple organizations participating in shared OA component development and cybersecurity assurance? What kinds of acquisition management practices and analysis tools are needed for the acquisition workforce to insure cost savings and BBP in such settings? Addressing these questions is beyond the scope of this paper, but these questions require follow-on acquisition research to resolve and answer.

Better Buying Power 2.0 Goals

Better Buying Power (http://bbp.dau.mil/) is part of DoD's mandate to *do more without more* by implementing best practices in acquisition. BBP identifies seven areas of focus that group a larger set of 36 initiatives that offer the potential to restore affordability in defense procurement and improve defense industry productivity. One of the seven areas focuses on promoting or increasing competition, and this area includes an initiative to "enforce open system architectures and effectively manage technical data rights" [DAU12]. Technical data rights pertain to two categories of Intellectual Property (IP): they refer to the Government's rights to (a) technical data (TD – e.g., product design data, computer databases, computer software documentation); and (b) computer software (CS – e.g., source code, executable code, design details, processes, and related materials). These rights are realized through IP licenses provided by system product or service providers (e.g., software producers) to the Government customer, so long as the customer fulfills the obligations stipulated in the license agreement (e.g., to indicate how many software users are authorized to use the licensed product or service according to a fee paid).

As already noted, our acquisition research has focused on issues addressing OA systems and IP licenses since 2008 [ScA08], as well as forward to the acquisition of secure OA systems for command and control (C2) and enterprise information systems [ScA11,ScA12b,ScA13b], where security requirements can be expressed in a manner similar to IP obligations and rights. Therefore, here we turn to identify how a sample of different goals of BBP 2.0 initiatives interact or relate to the trends and challenges examined so far in this paper. The BBP goals are highlighted, then followed by a brief examination.

- *Increase competition* – One central purpose for acquiring OA systems is to increase the likelihood of competition among system producers who can provide software components that can be replaced by similar offerings by other component producers. We demonstrate how this can work when system architectures are explicitly modeled, and their software components and interconnections are similarly specified in an open manner [AlA13, ScA12a].
- *Adopt OA systems that utilize standardized interfaces* – Open system architectures that can accommodate common components from alternative producers requires that the components utilize standardized interfaces, whether in the form of open Application Program Interfaces (APIs), standard data exchange protocols, and standard data representations, formats, and meta-data [ScA08]. But also noted earlier, app and widget components at present have a plethora of standardized interfaces, and it is unclear which will survive, be sustained, be widely adopted (inside/outside of DoD), and be evolved [End13a].
- *Utilize open source software components where appropriate to reduce costs* – another aspect of openness that OA systems embrace and DoD policy accepts is to utilize system components developed as open source software (OSS) [DIS12]. Utilization of OSS components, along with composing OA systems that incorporate OSS and closed, proprietary components, does require careful attention to the management and analysis of multiple IP licenses that apply to different OA system components, as well as determining what overall IP and/or cybersecurity rights and obligations cover the overall system [AlA13, ScA12a], especially for C2 systems [AlA13, ScA13b, ScA13c].
- *Increase small business roles and opportunities* – one way to increase competition in the realm of OA systems is to identify where smaller scale

software applications (apps) or widgets can be utilized, which might be produced by small businesses or startup ventures which dominate much of the online markets for Web-based or mobile device apps/widgets. Small businesses may further be advantaged by their utilization of OSS infrastructure components, platforms, or remote services, since large commercial contractors may not see sufficient profit margins to develop proprietary alternatives. So OA systems that accommodate OSS components that can integrate custom apps/widgets into innovative system capabilities (C2SC), may then realize new opportunities for DoD customers. Other small business opportunities may similarly arise for such ventures that focus on emerging cybersecurity assessment or tool development services.

- *Use technical development phase for true risk reduction and rapid prototyping* – In looking forward, there is potential interest in seeing the BPP initiative evolve to also address risk as an implicit cost driver. This might allow or innovative ways and means to reduce emerging risks through accelerated or "look ahead" system acquisition and development approaches that emphasize increased reliance on rapid prototyping. This kind of rapid prototyping might even be performed by appropriately trained end-users or warfighters. A move towards OA systems for Web-based and mobile devices that rely on apps/widgets retrieved from online marketplaces, that can be composed through interpretive software program "scripting" and mashup techniques, is a clear example of this [End13, GMH13 GuW12, ScA13a]. Thus, it is not surprising to find such emerging techniques being investigated and assessed for possible production of new C2 capabilities [GBC14, GMH13, ScA13b].
- *Do more without more* – an overall summary of the current BBP initiative is focusing attention of how to make acquisition more agile, to do more without more, and to develop a new generation acquisition workforce that can enact acquisition processes that are thin and flexible when needed, yet robust and cost-effective, while also being amenable to continuous improvement. This is indeed a real challenge to fulfill, and beyond the scope of what current acquisition practices are likely to achieve without targeted investment in acquisition improvement research. To be clear, one just needs to consider emerging opportunities (and potential asymmetric cybersecurity threats) that arise through the desire to develop next-generation C2SC that are to be composed from apps/widgets that can operate on Web-based/mobile devices. What are the best processes or practices for acquiring, developing, and sustaining deployed systems that are to be built using these new software technologies (e.g., apps/widgets for mobile devices)? How should these processes and practices be adapted to accommodate personal devices (e.g, Apple iPhones, Android tablet, Microsoft Mobile Phone, Blackberry 10 phone) that individual warfighters, joint force troops, or contracted service providers bring with them into the battlespace? How must acquisition processes be best adapted to accommodate and rely on software supply chains that arise around consumer-oriented app marketplaces as possible ways/means for

doing more (e.g., rapidly prototyping warfighter composable C2 app/widget mashups [GMH13]) *without more* (e.g., warfighters who bring their own mobile computing devices for use in C2 contexts) [GBC14]? Once again, these are critical questions to address and resolve through new acquisition research and supporting technology development.

Emerging Challenges in Achieving BBP through OA Software Systems

The business models and IP licenses for software components are tightly coupled: software component licenses codify component producer business models. Said more simply, licenses codify business models. So different software business models imply different software license obligations and rights, and different license types reflect different possible business models. Licenses are generally recognized as contracts regarding IP expressed through terms and conditions that specify obligations and rights stipulated by the component's producer to enable/constrain what can be done with the component by its integrator or end-users. Understanding and assuring software IP obligations and rights is iroutinely a task for acquisition management, and thus a task to be competently performed by the acquisition workforce

Obligations (like purchase costs/fees paid, or to insure access to open source software code modifications) denote conditions, events, or actions imposed by a software producer (the licensor) that must be fulfilled by the software integrator/customer enterprise (the licensee) in order to realize the *rights* identified in the licenses (right to use; right to distribute copies; no right to distribute modified copies, etc.). Note that software system integrators play a role is shaping the obligations and rights imposed on customer enterprises based on choices they make in how software component-based systems are designed, built, and deployed. So where/who does system integration occurs matters, as does whether customer enterprises that acquire systems have policies that determine which software licenses (or business models) they will accept.

Similarly, we note that "cybersecurity requirements" can also be expressed and analyzed in terms of obligations and rights [ScA11, ScA12b]. This suggests the the problems and solutions to software IP license management will be similar in kind or form to those for cybersecurity assurance. Below, we just focus attention to software IP obligations and rights, though the same consequences may apply to the cybersecurity of OA systems and components.

There are many unstated consequences that can arise when software licenses are not well understood. Here are some examples we have seen within the DoD context.

- *Different military services specify which software licenses they do and do not accept*. This can give rise to service **X** refusing to use any software component subject to license ***A*** (e.g., GPL—Gnu Public License), while service **Y** deploys mission-critical command and control systems that incorporate components subject to license ***A*** [ScA08]. This may imply that service **X** will not allow connection of its C2 systems to service **Y** C2 systems, which can readily look like a bad outcome.

- *Acquisition program managers/staff (including in-house legal counsel) may not understand how software licenses affect OA system design, and vice-versa.* Component-based system design can determine which software licenses will fit, or which can fit if the system design is altered to encapsulate desirable software components with somewhat problematic license obligations or rights [ScA13a].
- *Software license obligations and rights propagate through system development life cycle activities in ways not well understood by system developers, integrators, end-users, or acquisition managers*. We have investigated and described many examples of this in a recent paper that shows how license constraints are mediated by software system design, build-integration, deployment, post-deployment support tools and activities.
- *Different acquisition programs within DoD and other government agencies may independently reinterpret software component licenses.* This realizes enterprise-wide inefficiencies, as well as increases avoidable costs. It appears to be technically possible to codify software component licenses by type or producer, especially with regards to performative obligations and operational rights that Program Offices or customer organizations seek. The license modeling techniques we have investigated demonstrates the potential, practicality, and scalability of such possibility [AlA13, ScA12a, ScA12b, ScA13b]. However, it may be most efficient and most effective for DoD to have common legal interpretations for different licenses (or different business models). Such interpretations could be common, if produced by a central legal authority (e.g., Office of General Counsel). Alternatively, it may also be possible for DoD and other government agencies to provide an open framework or (acquisition) policy guidance whose purpose is to encourage software producers to not only provide software licenses in current narrative forms, but also to provide them in computer processable forms (using domain-specific languages) amenable to automated license analysis. Once again, this is a form of guidance and training we can provide, but it is not one that we can impose on anyone. We believe it is in the best interest of DoD and other government agencies to employ software licenses that are both human readable and formally processable though automated means, at least in terms of software license obligation and right determinations.
- *Failure to understand software license obligation and rights propagation can reduce DoD buying power, increase software life cycle costs, and reduce competition*. Guidance from the OUSD for Acquisition, Technology, and Logistics recommends programmatic adoption of different Better Buying Power initiatives grouped into seven focus areas of relevance (http://bbp.dau.mil/sevenareas.html) as programmatic methods for doing more without spending more. Acquiring licensed software components is a cost-generating activity, whose costs/fees can be reduced while acquiring evermore agile and adaptive software components and open architecture

component-based systems. However, software license non-compliance or worse, infringement, on the part of DoD will generate costs, program delays, as well as reduce agility and adaptation, all of which can be avoided. Such situations can and must be avoided through acquisition and development practices with little/no additional cost to affect. Such practices can be codified within open source business processes or open source computational business process models that can be shared, customized to specific program needs, redistributed and archived [ScA13b].

- *Software producers often provide idiosyncratic licenses that generally conform to common business models and common license types.* This seems mainly to arise from efforts by software producers to protect or update their business models in ways that improve their financial yield or protect/lock-in their customer base. This in turn generates demand for time, attention, and effort from legal counsel that support acquisition programs, while also reducing the effectiveness and timeliness of program acquisition efforts. DoD and other government agencies may be able to explicitly specify in advance what kinds of generic software license obligations they will accept and what kinds of generic software rights they seek, through their own explicit business models. Such specifications can be codified and provided to software producers in open source manner through software license acquisition policies. Software producers might then separate license terms and conditions that do and do not address current license acquisition policies, in order to streamline licensing design and analysis practices for the mutual benefit of software producers, integrators, and customers.
- *Software producers generally provide software licenses that are assumed to legally dominate in systems composed of components from different software producers or integrators.* We refer to software systems (or systems of systems) composed from components (e.g., apps, widgets) subject to different licenses as “heterogeneously-licensed systems” (HLS) [AlS10, AlA13]. Popular Web browsers that are compatible with widgets, apps, or plug-in components (e.g., Google Chrome, Mozilla Firefox) are subject to dozens of component licenses. Popular COTS software components also sometimes encompass components subject to multiple licenses. In both situations, the component producer asserts overall component license obligations and rights in ways that are compatible with the licenses included therein (or so we hope). But when we deploy components that are composed into complex system architectures, or employ components that support on-demand download and implicit integration of smaller components (widgets, plug-ins, scripts, etc.) from online stores, then analysis of license obligation and rights propagation or encapsulation matters. Such technical details can readily overwhelm program acquisition managers and legal staff, thereby reducing the agility and adaptation of component-based system development/deployment. Provision of automated license analysis capabilities within software license management systems should be able to overcome this situation.

- *Given the challenges of HLS, it is unclear what kinds of trade-offs can/should be addressed by software system integrators or program acquisition staff to maximize overall system development agility and evolutionary adaptation.* This situation is not unique to DoD, but is in fact widespread. However, as DoD and other government agencies move to embrace agile and adaptive component-based software systems to realize new, more timely system capabilities at lower cost compared to legacy approaches, then there is need to provide guidance for how to identify and manage such trade-offs. Failure to recognize the challenges of analyzing and managing HLS systems translates into opportunities lost while avoidable costs increase. We can and should do better than this. But this will require that resources be allocated to identify, articulate, train, and iteratively refine best practices about how, where, when, and why these trade-offs arise. Such knowledge should therefore be captured, codified, shared, accessed, updated, and redistributed in an open source manner.
- *Software IP license and cybersecurity obligations and rights must be tracked, accounted, and managed*. A move to component-based open architecture systems increases organizational overhead for managing software licenses. This overhead can be reduced, or better transformed into productive, value-adding business practices, through use of automated software obligations and rights management systems (SORMS). While SORMS exist and are routinely used by software component producers (to keep track of who has a licensed copy of their software products), SLMS do not exist at this time for software system integrators or customer enterprises.
- *DoD and other Government agencies would financially and administratively benefit from engaging the development and deployment of an open source automated SORMS.* This may represent the lowest cost means for simplifying license analysis while maximizing the benefits of agile and adaptive component-based software systems acquisition within the DoD and other government agencies. SORMS can help to better DoD software buying power. Similarly, an open source SORMS would also be of value to smaller or startup software producers who may best be able to create innovative and agile software components (widgets) in cost-competitive ways. Last, an open source SORMS intended for software integrator/customer enterprises would be of value to large, established DoD software producers, as a medium through which larger-scale software component acquisitions (e.g., components acquired for standardized deployment throughout an enterprise can be negotiated and simplified.

Finally, as suggested along the way, *all* of these consequences can be both anticipated and mitigated through action and careful investment in enabling solutions.

New Practices to Realize Cost-Effective Acquisition of OA Software Systems

The trends and concerns identified above point to substantial challenges in identifying what can be done to both realize cost-effective BBP, and to do so in ways that enable and empower the acquisition workforce in the years ahead. Technology, better buying practices, new business models, new cybersecurity requirements all point to the need for future research and development of new acquisition support technologies work processes, and guidance practices. The goal is to make sure that acquisition management time and effort does not become the main cost and the main risk factor going forward on the path to agile OA C2 system development, deployment, and sustaining system evolution.

At this point, we see at least three key areas of opportunity for future acquisition research and development.

First, we need to research and develop ***worked examples*** of well-formed OA system architectures that are appropriate for C2 system capabilities, and that accommodate Web-based apps, widgets and mobile devices. Such OA system architectures should specify representative and standardized component interfaces. The examples should also include carefully specified shared agreements that account for different IP licenses and diverse business models of software producers, system integrators, and multiple end-user organizations who must collectively act in ways that enable agile development and adaptive evolution of demonstrable C2 system capabilities.

Second, we need robust, ***open source models*** of application security processes and reusable cybersecurity requirements that account for exigencies in heterogeneous app/widget software ecosystems, account for software evolution dynamics, formation and continuous improvement of automatable shared agreements, and more. These models should account for description of current process practices, prescription of required verification and validation activities and outcome (deliverable documents or online artifacts), and proscription of what tools/techniques to use, by whom, when, where and how.

Third, we need precise ***domain specific languages*** (DSLs) for specifying, ***and automated analysis tools*** for continuously assessing and continuously improving, cybersecurity and IP license requirements for dynamically evolving Web/mobile C2 system-based capabilities. The DSLs needed must be able to specify and operationalize the shared agreements between different DoD organizations, government agencies and commercial enterprises involved in producing, integrating, or evolving component-based OA C2 system capabilities.

Overall, what we call for is similar in kind to what we have already produced and applied in other software development domains, using then current technologies [JeS05, SJN05, ScA08, SBN13]. What we now call for is a reinvention and repurposing of these concepts, but in contemporary forms scaled and secured in ways that best meet the needs of the DoD Program Offices, acquisition program managers, and others in the acquisition workforce.

Conclusions

Our presentation focused on our ongoing investigation and refinement of techniques for identifying and reducing the costs, streamlining the process, and improving the readiness of future workforce for the acquisition of complex software systems. Emphasis was directed at identifying, tracking, and analyzing software component costs and cost reduction opportunities within acquisition life cycle of open architecture (OA) command and control (C2) system capabilities. These systems are expected to combine best-of-breed software components and software products lines (SPLs) in the form of Web-based applications (apps) and user interface widgets that are subject to different intellectual property (IP) license and cybersecurity requirements.

Our study reported in this paper also identifies a new set of technical risks that can dilute the cost-effectiveness of Better Buying Power efforts. It similarly suggests that current acquisition practices aligned with BBP can also give rise to acquisition management activities that can dominate and overwhelm the costs of OA system development. This adverse condition can arise through app/widget vetting, new software business models, opaque and/or underspecified acquisition management processes, and the evolving interactions of new software development and deployment techniques. Unless proactive investment in acquisition research and development can give rise to worked examples, open source models, and new acquisition management system technologies, the likelihood of acquisition management dominating agile development and adaptive deployment of component-based OA C2 system capabilities.

The Department of Defense, other government agencies, and most large-scale business enterprises continually seek new ways to improve the functional capabilities of their software-intensive systems. The acquisition of OA systems that can adapt and evolve through replacement of functionally similar software component applications (apps) and widgets is an innovation that can lead to lower cost systems through more agile system development and adaptive system evolution. Our research identifies and analyzes how new software component apps and widgets, their IP license and cybersecurity requirements, and new software business models can interact to drive down (or drive up) total system costs across the system acquisition life cycle. The availability of such new scientific knowledge and technological practices can give rise to more effective expenditures of public funds and improve the effectiveness of future software-intensive systems used in government and industry.

Overall, this paper serves to help describe and detail how software component technologies, IP licenses, security requirements, business models and adaptive system evolution interact, as well as what policies, practices, or technologies within DoD and other government agencies can simplify or exacerbate OA system cost arising at different points in the acquisition life cycle. Our common goal is to increase the ways, means, and beneficial consequences of the transition to the cost-effective acquisition of component-based OA software systems whose acquisition, development, deployment, and ongoing evolution are agile and adaptive.

Acknowledgements
The research described in this report was supported by grant N00244-14-1-0030 from the Acquisition Research Program at the Naval Postgraduate *School*, Monterey, CA. No endorsement, review, or approval implied. This paper reflects the views and opinions of the authors, and not necessarily the views or positions of any other persons, group, enterprise, or government agency.

References

[AIA13] Alspaugh, T.A, Asuncion, H. and Scacchi, W. (2012). The Challenge of Heterogeneously Licensed Systems in Open Architecture Software Ecosystems, S. Jansen, S. Brinkkemper, and M. Cusumano (Eds.), *Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry,* Edward Elgar Publishing, 103-120, Northampton, MA.

[AIS10] Alspaugh, T.A, Scacchi, W., and Asuncion, H. (2010). Software Licenses in Context: The Challenge of Heterogeneously Licensed Systems, *Journal of the Association for Information Systems*, 11(11), 730-755, November 2010.

[CFG13] Chudnovsky, O., Fischer, C. Gaedke, M. and Pietschmann (2013). Inter-Widget Communication by Demonstration in User Interface Mashups. *Web Engineering*, Springer-Verlag, Lecture Notes in Computer Science, Vol. 7977, 502-505.

[DAG14] *Defense Acquisition Guidebook* (2014). CVE--Common Vulnerabilities and Exposures. Chapter 13.7.3.1.4, https://acc.dau.mil/CommunityBrowser.aspx?id=492079#13.7.3.1.4 accessed April 2014.

[DAU12], Defense Acquisition University (2012). *Open Systems Architecture and Technical Data Rights...Management Approaches*, http://bbp.dau.mil/docs/Open%20Systems%20Architecture%20and%20Technical%20Data%20Rights%20.%20.%20.%20Management%20Approaches.pdf accessed 30 October 2012.

[DIS12], Defense Information Systems Agency (2012). *DOD Open Source and Community Source Software Development in Forge.mil*, SoftwareForge Document ID – doc26066doc26066 http://www.disa.mil/News/Conferences-and-Events/DISA- Mission-Partner-Conference-2012/~/media/Files/DISA/News/Conference/2012/ DoD_Open_Source_Community_Forge.pdf accessed 30 October 2012.

[DoD12] Department of Defense (2012). *Joint Operational Access Concept*, Version 1.0, 17 January 2012, http://www.defense.gov/pubs/pdfs/JOAC_Jan%202012_Signed.pdf

[DoDOSA11], Department of Defense Open Systems Architecture (2011). *Contract Guidebook for Program Managers*, Vol. 0.1, December, https://acc.dau.mil/OSAGuidebook

[End13] Endres-Niggemeyer, B. (2013). The Mashup Ecosystem, in *Semantic Mashups: Intelligence Reuse of Web Resources*, Springer, 1-50.

[End13a] Endres-Niggemeyer, B. (2013). Mashups Live on Standards, in *Semantic Mashups: Intelligence Reuse of Web Resources*, Springer, 51-89.

[GBC14] George, A., Bowers, A., Galdorisi, G., Hszieh, S., Morris, M., and Raney, C. (2014). DoD Application Store: Enabling C2 Agility, *Proc. 19Th Intern. Command and Control Research and Technology Symposium*, Paper-104, Alexandria, VA, June 2014.

[GMH13] George, A., Morris, M., Galdorisi, G., Raney, C., Bowers, A., and Yetman, C. (2013) Mission Composable C3 in DIL Information Environments using Widgets and App Stores. *Proc. 18Th Intern. Command and Control Research and Technology Symposium*, Paper-036, Alexandria, VA, June 2013.

[GuW12] Guertin, N. and Womble, B. (2012). Competition and the DoD Marketplace, *Proc. 9th Acquisition Research Symposium*. Vol. 1, 76-82, Naval Postgraduate School, Monterey, CA.

[Giz11] Gizzi, N. (2011). Command and Control Rapid Prototyping Continuum (C2RPC) Transition: Bridging the Valley of Death, *Proceedings 8th Annual Acquisition Research Symposium,* Vol. 1, Naval Postgraduate School, Monterey.

[Han13] Hanf, D. (2013). *MPE/AAE Business Model Framework Overview*. Mitre Corporation, personal communication, July 2013.

[JeS05] Jensen, C. and Scacchi, W. (2005). Process Modeling Across the Web Information Infrastructure, *Software Process--Improvement and Practice,* 10(3), 255-272, July-September 2005.

[ReB12] Reed, H., Benito, P., Collens, J., and Stein, F. (2012). Supporting Agile C2 with an Agile and Adaptive IT Ecosystem, *Proc. 17th Intern. Command and Control Research and Technology Symposium* (ICCRTS), Paper-044, Fairfax, VA, June 2012.

[ScA08] Scacchi, W. and Alspaugh, T., (2008). Emerging Issues in the Acquisition of Open Source Software within the U.S. Department of Defense, *Proc. 5th Acquisition Research Symposium*, NPS-AM-08-036, Naval Postgraduate School, Monterey, CA, May.

[ScA11] Scacchi, W. and Alspaugh, T., (2011). Advances in the Acquisition of Secure Systems Based on Open Architectures, *Proc. 8th Acquisition Research Symposium*, Vol. 1, Naval Postgraduate School, Monterey, CA.

[ScA12a] Scacchi, W. and Alspaugh, T., (2012a) Understanding the Role of Licenses and Evolution in Open Architecture Software Ecosystems, *Journal of Systems and Software*, 85(7), 1479-1494, July 2012.

[ScA12b] Scacchi, W. and Alspaugh, T., (2012b). Addressing Challenges in the Acquisition of Secure Software Systems with Open Architectures, *Proc. 9th Acquisition Research Symposium*, Vol. 1, 165-184, Naval Postgraduate School, Monterey, CA.

[ScA13a] Scacchi, W. and Alspaugh, T. (2013a). Processes in Securing Open Architecture Software Systems, *Proc. 2013 Intern. Conf. Software and System Processes*, San Francisco, CA, May 2013.

[ScA13b] Scacchi, W. and Alspaugh, T.A. (2013b). Streamlining the Process of Acquiring Secure Open Architecture Software Systems, *Proc. 10th Annual Acquisition Research Symposium*, Monterey, CA, 608-623, May 2013.

[ScA13c] Scacchi, W. and Alspaugh, T.A. (2013c). Challenges in the Development and Evolution of Secure Open Architecture Command and Control Systems, *Proc. 18Th Intern. Command and Control Research and Technology Symposium*, Paper-098, Alexandria, VA, June 2013.

[SBN12] Scacchi, W., Brown, C. and Nies, K. (2012). Exploring the Potential of Virtual Worlds for Decentralized Command and Control, *Proc. 17th Intern. Command and Control Research and Technology Symposium* (ICCRTS), Paper 096, Fairfax, VA, June 2012.

[SJN06] Scacchi, W. Jensen, C., Noll, J. and Elliott, M. (2006). Multi-Modal Modeling, Analysis, and Validation of Open Source Software Development Processes, *Intern. J. Internet Technology and Web Engineering,*1(3), 49-63, 2006.

[Wei14] Weir, M. (2014). BYOD Topic: How Complicated Can Calendars Be? *J. Cybersecurity and Information Systems*, 2(1). 18-19.

[YRM13] Yetman, C., Raney, C., Morris, M. and George, A. (2013) UxV Data to the Cloud via Widgets, *Proc. 18Th Intern. Command and Control Research and Technology Symposium*, Paper-051, Alexandria, VA, June 2013.