SYM-AM-15-088

PROCEEDINGS
OF THE
TWELFTH ANNUAL ACQUISITION
RESEARCH SYMPOSIUM

THURSDAY SESSIONS
VOLUME II

Achieving Better Buying Power Through Acquisition of Open Architecture Software Systems for Web-Based and Mobile Devices

Walt Scacchi, University of California–Irvine
Thomas Alspaugh, University of California–Irvine

Published April 30, 2015

Disclaimer: The views represented in this report are those of the author and do not reflect the official policy position of the Navy, the Department of Defense, or the federal government.

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

The research presented in this report was supported by the Acquisition Research Program of the Graduate School of Business & Public Policy at the Naval Postgraduate School.

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

Achieving Better Buying Power Through Acquisition of Open Architecture Software Systems for Web-Based and Mobile Devices[1]

Walt Scacchi—is senior research scientist and research faculty member at the Institute for Software Research, University of California, Irvine. He received a PhD 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 International Conference on Software Engineering—Practice Track, and in 2012, he served as general co-chair of the 8th IFIP International Conference on Open Source Systems. [wscacchi@ics.uci.edu]

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 PhD, 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. [alspaugh@ics.uci.edu]

Abstract

Many people within large enterprises rely on up to four Web-based or mobile devices for their daily work routines—personal computer, tablet, and personal and work-specific smartphones. Our research is directed at identifying, tracking, and analyzing software component costs and cost reduction opportunities within the acquisition life cycle of open architecture (OA) systems for such Web-based and mobile devices. These systems are subject to different intellectual property license and cybersecurity requirements. Our research goal is to create a new approach to address challenges in the acquisition of software systems for Web-based or mobile devices used within academic, business, or government enterprises. Acquisition personnel in such enterprises will increasingly be called on to review and approve choices between functionally similar open source software (OSS) components, and commercially priced closed source software (CSS) components, to be used in the design, implementation, deployment, and evolution of secure OA systems. We seek to make this a simpler, more transparent, and more tractable process. Finally, this acquisition research supports and advances a public purpose by investigating acquisition challenges arising from the adoption and deployment of secure OA software systems for Web-based or mobile devices.

Overview

The Department of Defense (DoD), 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

[1] This report was supported by 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.

thus seen as an approach to realizing Better Buying Power (BPP) goals for lowering system costs, achieving technical excellence, enabling innovation, and advancing the acquisition workforce.

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 (standalone and plug-in "apps") and widgets that exploit modern Web capabilities. We then turn to examine some key goals of the BBP 3.0 initiative (Kendall, 2014) 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; and (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 (Agre et al., 2014; George, Bowers, et al., 2014; George et al., 2013; Guertin & Womble, 2012; Reed et al., 2012; Scacchi & Alspaugh, 2012b, 2013c, 2014a). 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 "assembled capabilities for C2 systems" (AC-C2). 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. We also focus on design of OA system capability involving office productivity and social media components (Agre et al., 2014) that increasingly may be configured within a secure AC-C2 (Scacchi & Alspaugh, 2011, 2012b, 2013b).

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 AC-C2s or C2 system elements (Reed et al., 2012; Reed et al., 2014; Scacchi &

Alspaugh, 2014a, 2014b, 2014c). 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 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, via online storefronts (George, Bowers, et al., 2014; George, Galdorisi, et al., 2014; George et al., 2013).

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 (Agre et al., 2014; Gizzi, 2011; George et al., 2013; Scacchi & Alspaugh, 2013b). Widgets can provide summary information or a limited view into a larger application that can be used alongside related widgets to provide 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 mismatches 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 (DoD, 2012; Gizzi, 2011).

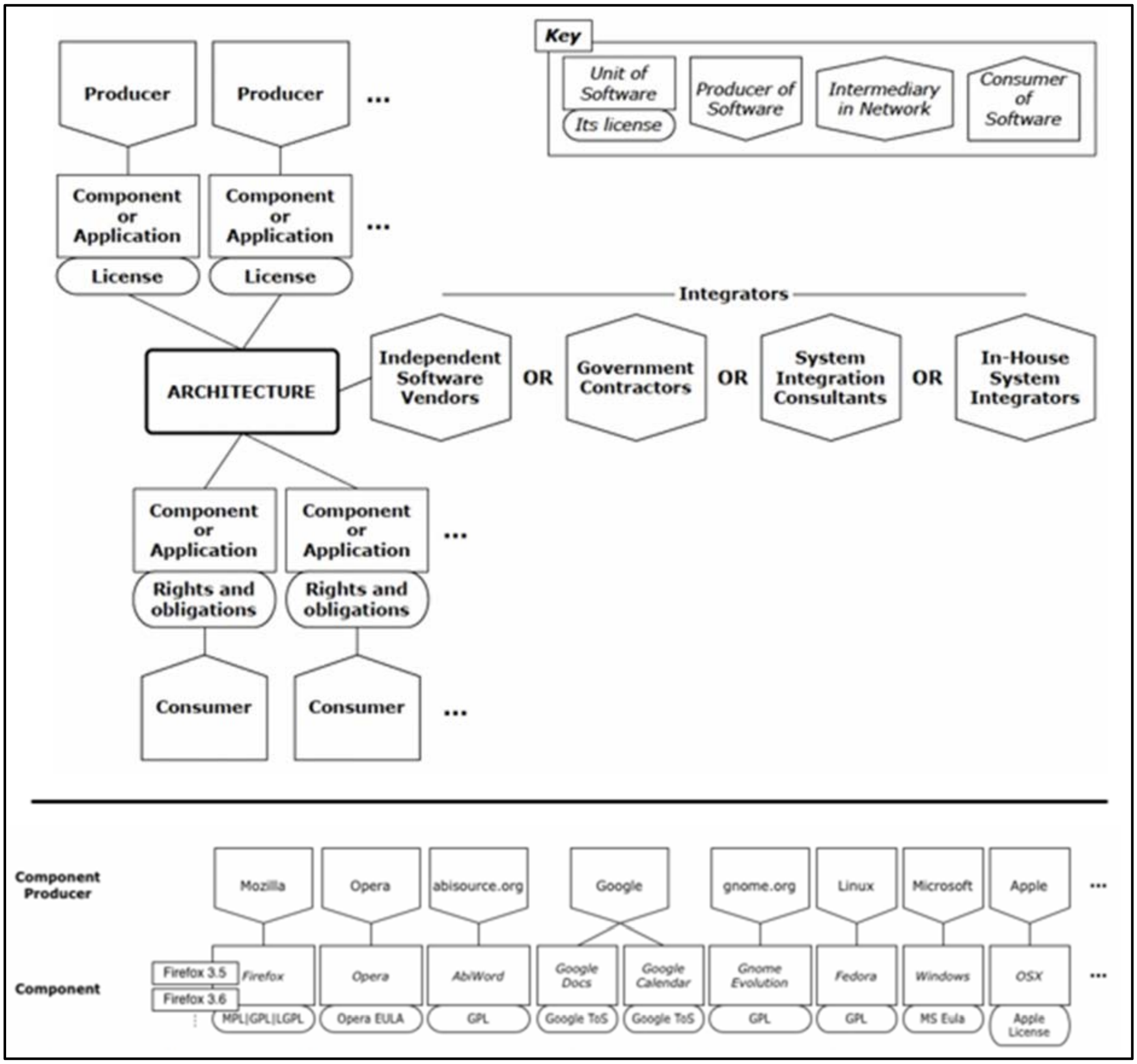


Figure 1. A Generic Software Ecosystem Supply Network (Upper Part), Along With a Sample Elaboration of Producers, Software Component Applications, and Licenses for OA System Components They Employ (Lower Part)

(Scacchi & Alspaugh, 2012a)

This decentralization will engender acquisition and development of heterogeneously-licensed systems (HLS), whereby different software components (apps, widgets) will be subject to different IP licenses (Alspaugh et al., 2012; Alspaugh et al., 2010), as well as to different cybersecurity requirements (Defense Acquisition Guidebook, 2014; Scacchi & Alspaugh, 2012b, 2013a, 2013b, 2013c). This 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 (Scacchi & Alspaugh, 2012a, 2013b). These 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 (George, Bowers, et al., 2014), and thus potentially mitigate the benefits of BBP that can arise from MPE and AAE for C2 systems.

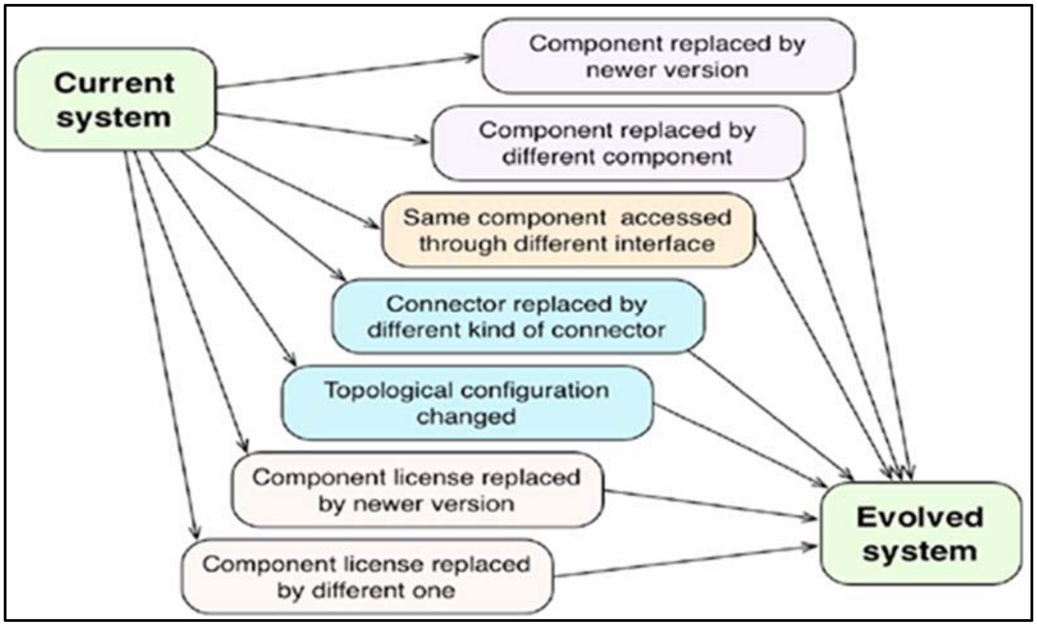

Current system
Component replaced by newer version
Component replaced by different component
Same component accessed through different interface
Connector replaced by different kind of connector
Topological configuration changed
Component license replaced by newer version
Component license replaced by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

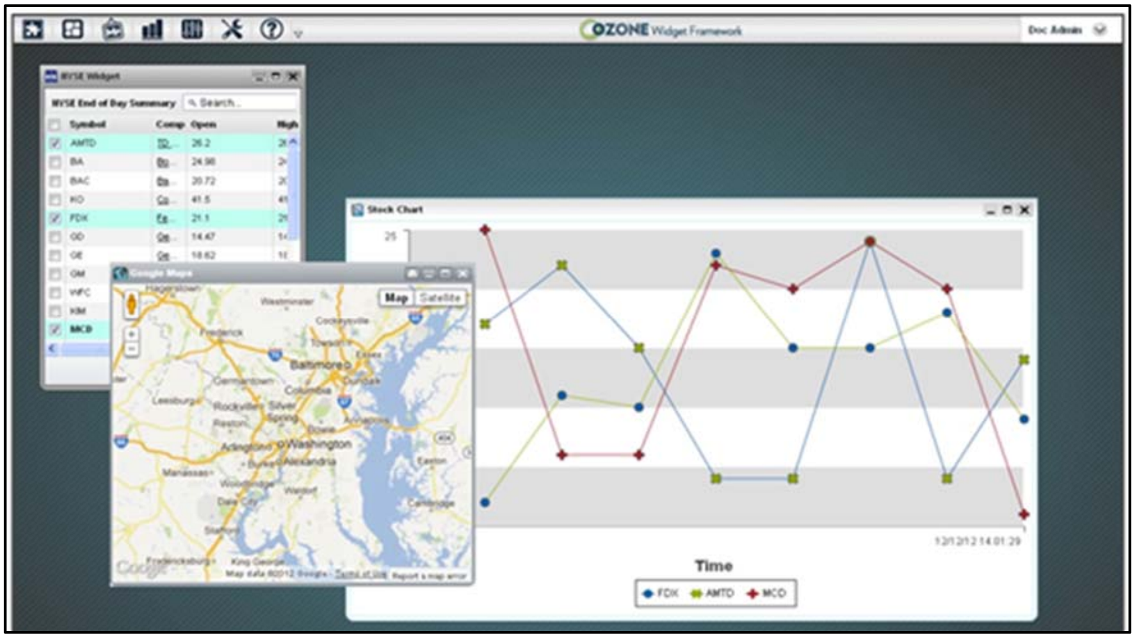
Figure 2. The Kinds of Common Evolutionary Changes That Arise During OA Software Component Development, Deployment, and Sustained Usage

Moving Towards Shared Development of Apps and Widgets as OA System Components

Future OA systems for agile C2 may be 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) is 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 (Scacchi & Alspaugh, 2013b). 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 enable 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.

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.



NYSE End of Day Summary

☐	Symbol	Comp	Open
☑	AMTD	TD...	26.2
☐	BA	Bo...	24.98
☐	BAC	Ba...	20.72
☐	KO	Co...	41.5
☑	FDX	Fe...	21.1
☐	GD	Ge...	14.47
☐	GE	Ge...	18.62
☐	GM		
☐	WFC		
☐	XIM		
☑	MCD		

Figure 3. OWF Widgets Running Within a Web Browser

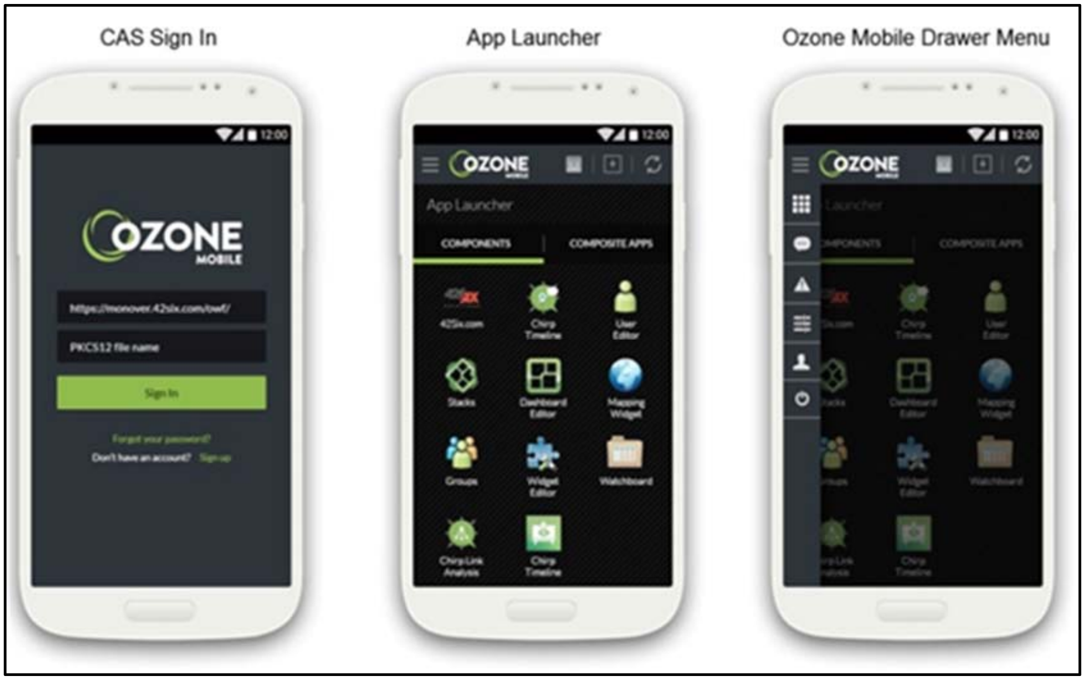


Figure 4. OWF Widgets Running on a Mobile Device

BYOD suggests that end-users and warfighters are bringing their own mobile devices with them into the field to support their missions. However, BYOD clearly exacerbates the technical challenges of cybersecurity assurance, often in ways that cannot be readily anticipated, as when independently developed components co-evolve in conflict to one another (Weir, 2014). Nonetheless, acquisition policy necessitates that cybersecurity vulnerability and exposures be addressed (*Defense Acquisition Guidebook*, 2015). 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 through an application security process for “vetting” these components.

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 the vetting can be performed by a centralized authority, such as by the operator of the Ozone Marketplace. But it is not clear that there will ever be only 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. Consider the example, seen in Figure 5, of a widget for “emergency response incident command system,” developed for the Department of Homeland Security (Rockwell, 2015). How do its components (possibly GOSS) compare or interoperate with widgets/AC-C2 from DoD agencies or program offices concerned with C2 system interoperability or AC-C2?

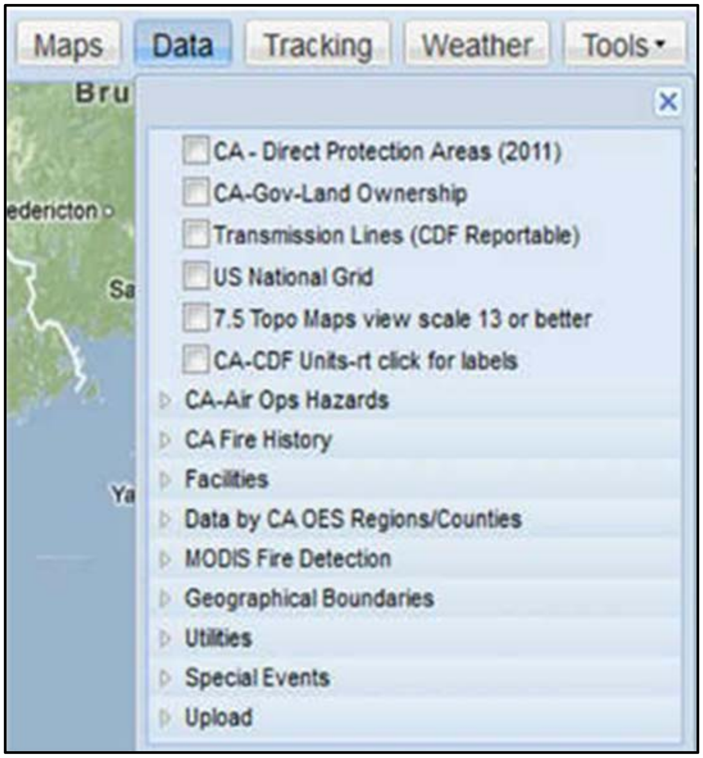


Figure 5. AC-C2 Style Widget From the Next-Generation Incident Response System for DHS

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 (Chudnovsky et al., 2013; Endres-Niggemeyer, 2013a). While OA systems may rely on “open standards”–style widget interfaces and communications patterns may 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 (Endres-Niggemeyer, 2013b).

Similarly, knowledge about the proper usage of widget components 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 mismatches), 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, and so forth. These are still new technical problems that are yet to be readily resolved or to have development/usage guides.

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 (Alspaugh et al., 2012; Scacchi & Alspaugh, 2011, 2012b, 2013b), 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) the following: (1) franchising, (2) enterprise licensing, (3) metered usage, (4) advertising supported, (5) subscription, (6) free component, (7) paid service/support fees, (8) federation reciprocity for shared development, (9) collaborative buying, (10) donation, (11) sponsored development, (12) free/open source software (e.g., Government OSS [GOSS]), and others (D. Hanf, personal communication, July 2013). Further, this list is not exhaustive; instead, it is only representative.

In contrast, for end-user organizations involved in agile development of OA system components, or an integrated system capability, there is a need to develop 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 (Scacchi & Alspaugh, 2013a). 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 (Scacchi & Alspaugh, 2013b, 2013c).

Software acquisition costs easily become difficult to predict/manage given the 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 (Scacchi & Alspaugh, 2012a, 2013a).

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 ensure 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 3.0 Goals

Better Buying Power (http://bbp.dau.mil/) is part of the DoD’s initiative that sees continuous improvement as the best approach to improving the performance of the defense acquisition enterprise. BBP 3.0 (Kendall, 2014) identifies eight areas of focus that group a larger set of itemized initiatives that offer the potential to restore affordability and realize technical excellence in defense procurement and improve defense industry productivity. One of the eight areas focuses on promoting or increasing competition, and this area includes an initiative to utilize modular open system architectures to stimulate innovation (Kendall, 2014). Technical innovations are constrained by two categories of Intellectual Property (IP) rights available to the government: (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 (Scacchi & Alspaugh, 2008), as well as forward to the acquisition of secure OA systems for command and control (C2) and enterprise information systems (Scacchi & Alspaugh, 2011, 2012b, 2013b), 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 3.0 initiatives interact or relate to the trends and challenges examined so far in this paper. The BBP goals are highlighted, and then followed by a brief examination.

- *Promote effective competition*—One central purpose for acquiring OA systems is to increase the likelihood of creating and maintaining competitive environments 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 (Alspaugh et al., 2012; Scacchi & Alspaugh, 2012a). Such openness also supports improved technology search and outreach, but enables retrieval of compatible OA system components from online (software app) storefronts.

- *Use Modular Open Systems Architecture to stimulate innovation*—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, as well as utilization of open source software (OSS) components (Scacchi & Alspaugh, 2008). But as 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 the DoD), and be evolved (Endres-Niggemeyer, 2013b).
- *Increase small business participation 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 innovative 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 shared 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 (AC-C2), 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.
- *Improve our leaders' ability to understand and mitigate technical risk*—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 for 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.
- *Increase the use of prototyping and experimentation*—The rapid development of Web-based or mobile app mashups might be performed by appropriately trained end-users or warfighters (Agre et al., 2014; Endres-Niggemeyer, 2013a). A move towards OA systems for Web-based and mobile devices that rely on apps/widgets retrieved from online marketplaces—apps composed through interpretive software program "scripting" and mashup techniques—is a clear example of this (Endres-Niggemeyer, 2013a; George et al., 2013; Guertin & Womble, 2012; Scacchi & Alspaugh, 2013a). Thus, it is not surprising to find such emerging techniques being investigated and assessed for possible production of new C2 capabilities (George, Bowers, et al., 2014; George et al., 2013; Scacchi & Alspaugh, 2013b).
- *Achieve dominant capabilities through innovation and technical excellence*—An overall summary of the current BBP initiative is focusing attention on how to make acquisition more agile, more innovative, and to develop a new generation acquisition workforce that can enact acquisition processes that are technically excellent—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 AC-C2 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/iPads, Android phones/tablets, Blackberry 10 phones) 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 [George et al., 2013]) without more (e.g., warfighters who bring their own mobile computing devices for use in C2 contexts; Agre et al., 2014; George, Bowers, et al., 2014)? 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 for Web-Based and Mobile Devices

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 routinely a task for acquisition offices, and thus a task to be competently performed by the acquisition workforce.

Obligations (like purchase costs/fees paid, or to ensure 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 in 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 system integration occurs and who does it 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 (Scacchi & Alspaugh, 2011, 2012b). This suggests that the problems and solutions to software component IP license management will be similar in kind or form to those for cybersecurity assurance. Below, we just focus attention on 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:

- *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 (Scacchi & Alspaugh, 2013a).
- *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 the 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 demonstrate the potential, practicality, and scalability of such possibility (Alspaugh et al., 2012; Scacchi & Alspaugh, 2012a, 2012b, 2013b). However, it may be most efficient and most effective for the 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 the 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 the 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.
- *Failures 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 BBP 3.0 initiatives grouped into eight focus areas of relevance as methods for innovation, continuous improvements, and doing more without spending more. Acquiring licensed software components is a cost-generating activity, whose costs/fees can be reduced while acquiring ever more agile and adaptive software components and open architecture component-based systems. However, software license non-compliance or worse, infringement, on the part of the DoD will generate costs, cause 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 (Scacchi & Alspaugh, 2013b).

- *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. The 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; Alspaugh et al., 2010; Alspaugh et al., 2012). Popular Web browsers that are compatible with widgets, apps, or plug-in components (e.g., Google Chrome, Mozilla Firefox, Apple Safari) are subject to dozens of component licenses. Popular commercial off-the-shelf (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 software system integrators or program acquisition staff make in order to maximize overall system development agility and evolutionary adaptation address*. This situation is not unique to the DoD, but is in fact widespread. However, as the DoD and other government agencies move to embrace agile and adaptive component-based software systems to realize new, more timely system capabilities at a lower cost compared to legacy approaches, then

there is a 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 the 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), SORMS do not exist at this time for software system integrators or customer enterprises.
- *The 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.
- *How best to cultivate and sustain DoD online storefronts and software ecosystem*. The acquisition of development of some DoD Web/mobile widgets may be strongly influenced by commercially available apps that are not secure, nor DoD information assured. Warfighters and others are often drawn to the best available technologies, including apps found in commercial online stores. Who decides whether apps in these conditions should be migrated, secured, and assured to meet DoD requirements? Alternatively, allowing such apps to be used as widgets for rapid prototyping new DoD AC-C2 may represent a promising new direction to stimulate innovation. Subsequently, this entails the needs to better understand possible commercial–DoD online storefront interactions and interdependencies, as well as articulating the needs of DoD agency/program office–specific storefronts. Next, we expect to see redundant app offerings across multiple storefronts, including challenges of identifying common apps of different versions or variants across storefronts and user devices (e.g., is Google Maps the same version across all platforms in use; is Apple Maps equivalent to Google Maps; is Google Earth compatible with NASA World Wind?). How best to determine when redundancy is good/bad for such apps/widgets is unclear and under-explored at this time. Last, as noted, software component

apps/widget licenses and business models across the DoD Software App Ecosystem are very diverse with unclear/unknown interactions and interdependencies. Business models are codified in Web/mobile app IP licenses (e.g., conferring right to use or EULAs) and cybersecurity requirements. Again, much remains here to investigate and resolve to best enable BBP 3.0 initiatives realized with Web-based and mobile software.

Finally, as suggested along the way, all of these consequences can be both anticipated and mitigated through action and careful investment that best enable BBP 3.0 compatible solutions.

New Practices to Realize Cost-Effective Acquisition of OA Software Systems for Web-Based and Mobile Devices

The trends and concerns identified above point to substantial challenges in identifying what can be done to both realize cost-effective BBP for Web-based and mobile device software apps, and to do so in ways that enable and empower the acquisition workforce in the years ahead. Technology, better buying practices, new business models, and 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 time and effort does not become the main cost and the main risk factor going forward on the path to agile OA Web-based or mobile compatible 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 automation-compatible 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 (Jensen & Scacchi, 2005; Scacchi & Alspaugh, 2008). 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 to best support BBP 3.0 initiatives for Web-based and mobile device software components (widgets, apps, plug-ins).

Conclusions

The DoD, 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.

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.

Overall, this paper serves to help describe and detail how Web-based and mobile device software component technologies, IP licenses, security requirements, business models, and adaptive system evolution interact. It also highlights what policies, practices, or technologies within the 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 Web-based and mobile device OA software systems whose acquisition, development, deployment, and ongoing evolution are agile and adaptive.

References

Agre, J. R., Gordon, K. D., & Vasiliou, M. S. (2014). Practical considerations for use of mobile apps at the tactical edge (Paper-035). In *Proceedings of the 19th International Command and Control Research and Technology Symposium (ICCRTS)*, Fairfax, VA.

Alspaugh, T. A., Asuncion, H., & Scacchi, W. (2012). The challenge of heterogeneously licensed systems in open architecture software ecosystems. In S. Jansen, S. Brinkkemper, & M. Cusumano (Eds.), *Software ecosystems: Analyzing and managing business networks in the software industry* (pp. 103–120). Northampton, MA: Edward Elgar.

Alspaugh, T. A., Scacchi, W., & Asuncion, H. (2010, November). Software licenses in context: The challenge of heterogeneously licensed systems. *Journal of the Association for Information Systems, 11*(11), 730–755.

Chudnovsky, O., Fischer, C., Gaedke, M., & Pietschmann, S. (2013). Inter-widget communication by demonstration in user interface mashups. *Lecture Notes in Computer Science, 7977,* 502–505.

Defense acquisition guidebook. (2014). CVE—Common vulnerabilities and exposures. Chapter 13.7.3.1.4. Retrieved April 2015 from https://acc.dau.mil/CommunityBrowser.aspx?id=492079#13.7.3.1.4

DoD. (2012, January 17). *Joint operational access concept*, Version 1.0. Retrieved from http://www.defense.gov/pubs/pdfs/JOAC_Jan%202012_Signed.pdf

Endres-Niggemeyer, B. (2013a). The mashup ecosystem. In *Semantic mashups: Intelligence reuse of web resources* (pp. 1–50). Springer.

Endres-Niggemeyer, B. (2013b). Mashups live on standards. In S*emantic mashups: Intelligence reuse of web resources* (pp. 51–89). Springer.

George, A., Bowers, A., Galdorisi, G., Hszieh, S., Morris, M., & Raney, C. (2014). DoD application store: Enabling C2 agility (Paper-104). In *Proceedings of the 19th International Command and Control Research and Technology Symposium*, Alexandria, VA.

George, A., Galdorisi, G., Morris, M., & O'Neil, M. (2014). DoD Application store: Enabling C2 agility? (Paper-104). In *Proceedings of the 19th International Command and Control Research and Technology Symposium (ICCRTS)*, Fairfax, VA.

George, A., Morris, M., Galdorisi, G., Raney, C., Bowers, A., & Yetman, C. (2013). Mission composable C3 in DIL information environments using widgets and app stores (Paper-036). In *Proceedings of the 18th International Command and Control Research and Technology Symposium*, Alexandria, VA.

Gizzi, N. (2011). Command and control rapid prototyping continuum (C2RPC) transition: Bridging the valley of death. In *Proceedings of the Eighth Annual Acquisition Research Symposium*, Vol. 1. Monterey, CA: Naval Postgraduate School.

Guertin, N., & Womble, B. (2012). Competition and the DoD marketplace. In *Proceedings of the Ninth Annual Acquisition Research Symposium*, Vol. 1 (pp. 76–82). Monterey, CA: Naval Postgraduate School.

Hanf, D. (2013, July). *MPE/AAE business model framework overview*. [Personal communication]. MITRE Corporation.

Jensen, C., & Scacchi, W. (2005, July–September). Process modeling across the web information infrastructure. *Software Process: Improvement and Practice, 10*(3), 255–272.

Kendall, F. (2014). *Better Buying Power 3.0 interim release*. Retrieved from http://www.acq.osd.mil/dpap/sa/Policies/docs/BBP_3_0_InterimReleaseMaterials.pdf

Reed, H., Benito, P., Collens, J., & Stein, F. (2012). Supporting Agile C2 with an agile and adaptive IT ecosystem (Paper-044). In *Proceedings of the 17th International Command and Control Research and Technology Symposium (ICCRTS)*, Fairfax, VA.

Reed, H., Nankervis, J., Cochran, J., Parekh, R., Stein. F., et al. (2014). Agile and adaptive ecosystem, results, outlook and recommendations (Paper-011). In *Proceedings of the 19th International Command and Control Research and Technology Symposium (ICCRTS)*, Fairfax, VA.

Rockwell, D. (2015, April 3). DHS transfer emergency-response tech. *Federal Computer Week.* Retrieved from http://fcw.com/articles/2015/04/03/dhs-nics.aspx

Scacchi, W., & Alspaugh, T. (2008). Emerging issues in the acquisition of open source software within the U.S. Department of Defense (NPS-AM-08-036). In *Proceedings of the Fifth Acquisition Research Symposium.* Monterey, CA: Naval Postgraduate School.

Scacchi, W., & Alspaugh, T. (2011). Advances in the acquisition of secure systems based on open architectures. In *Proceedings of the Eighth Annual Acquisition Research Symposium,* Vol. 1. Monterey, CA: Naval Postgraduate School.

Scacchi, W., & Alspaugh, T. (2012a, July). Understanding the role of licenses and evolution in open architecture software ecosystems. *Journal of Systems and Software, 85*(7), 1479–1494.

Scacchi, W., & Alspaugh, T. (2012b). Addressing challenges in the acquisition of secure software systems with open architectures. In *Proceedings of the Ninth Annual Acquisition Research Symposium,* Vol. 1 (pp. 165–184). Monterey, CA: Naval Postgraduate School.

Scacchi, W., & Alspaugh, T. (2013a). Processes in securing open architecture software systems. In *Proceedings of the 2013 International Conference on Software and System Processes*, San Francisco, CA.

Scacchi, W., & Alspaugh, T. A. (2013b). Streamlining the process of acquiring secure open architecture software systems. In *Proceedings of the 10th Annual Acquisition Research Symposium* (pp. 608–623). Monterey, CA: Naval Postgraduate School.

Scacchi, W., & Alspaugh, T. A. (2013c). Challenges in the development and evolution of secure open architecture command and control systems (Paper-098). In *Proceedings of the 18th International Command and Control Research and Technology Symposium,* Alexandria, VA.

Scacchi, W., & Alspaugh, T. (2014a). Achieving Better Buying Power through cost-sensitive acquisition of open architecture software systems (NPS-AM-14-C11P07R01-036). In *Proceedings of the 11th Annual Acquisition Research Symposium*. Monterey, CA: Naval Postgraduate School.

Scacchi, W., & Alspaugh, T. (2014b, August 19). *Cost-sensitive acquisition of open architecture software systems for mobile devices*. Invited presentation at the MITRE-ATARC Workshop on Challenges in Legal and Acquisition, Federal Mobile Computing Summit, Washington, DC.

Scacchi, W., & Alspaugh, T. (2014c, August 20). *Reasoning about the security of open architecture software systems for mobile devices*. Invited presentation at the Federal Mobile Computing Summit, Washington.

Weir, M. (2014). BYOD topic: How complicated can calendars be? *Journal of Cybersecurity and Information Systems, 2*(1), 18–19.

Acknowledgements

The research described in this report was supported by grants N00244-14-1-0030 and N00244-15-1-0010 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.

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

www.acquisitionresearch.net