

A Plausible Approach to Open Source Development in a Department of Defense (DoD) Environment: Open Technology Development (OTD)

Meggan Schoenberg
U.S. Navy, NAVSEA
Suffolk, VA

meggan.schoenberg@jcom.mil

Warren Bizub
Joint and Coalition Warfighting
Suffolk, VA

warren.bizub@jcom.mil

ABSTRACT

“Open Source” has made its way into DOD, but DOD is not capitalizing on its greatest values. DOD is currently exploring use of Open Source tools, services, and code but not so much on development philosophies and methodologies. In fact, many in DoD think it is just too hard. Why? Consider this definition of open source software development:

“Open source doesn’t just mean access to the source code. The distribution terms of open-source software must comply with the following criteria:

1. Free redistribution
2. Program must include source code and must allow distribution in source code as well as compiled form.
3. License must allow derived works
4. Integrity of the author’s source code
5. No discrimination against persons or groups
6. No discriminations against fields of endeavor
7. The rights attached to the program must apply to all whom the program is redistributed
8. License must not restrict other software
9. License must be technology-neutral”¹

This is a departure from DoD acquisition practices. It is generally accepted and encouraged to use proven products of open source, but not to develop DoD products in this manner. The industry proven benefits include reduced cost, opening the aperture of developers contributing to solutions, higher quality from community reviews, and accelerated technology innovation to name a few. DoD has tried in limited ways to open the door by hosting sites like Forge.mil, but that is not enough. This paper will make a case for expanding DoD’s use of open source development philosophies and methodologies. It will propose approaches to include changes in how we acquire software that could kick-start a revolutionary approach to efficient software development and rapid technology capability growth.

ABOUT THE AUTHORS

Mrs. Meggan Schoenberg is a Scientist for the Navy Combat Direction Systems Activity and the Joint National Training Capability (JNTC) Program Information Officer at the Joint and Coalition Warfighting Center. She has 20 years of experience developing and managing C4ISR, Modeling and Simulation (M&S), and business automation software systems. Mrs. Schoenberg holds BS degrees in both Mathematics and Computer Science, and an MS in Systems and Industrial Engineering, Engineering Administration.

Warren Bizub is the Program Director, Joint Advanced Concepts for the Joint and Coalition Warfighting Center. He served as the Technical Management Division Chief of the Joint Training Development Solutions Group and Director, Joint Advanced Training Technologies Laboratory at USJFCOM. Mr. Bizub holds a BS in Ocean Engineering, is a graduate of the Navy’s Senior Executive Management Development Program, and is a MIT Fellow in Foreign Politics, International Relations, and The National Interest.

¹ http://en.wikipedia.org/wiki/Open-source_software

A Plausible Approach to Open Source Development in a Department of Defense (DoD) Environment: Open Technology Development (OTD)

Meggan Schoenberg
U.S. Navy, NAVSEA
Suffolk, VA

meggan.schoenberg@jfc.com.mil

Warren Bizub
U.S. Joint Forces Command
Suffolk, VA

warren.bizub@jfc.com.mil

The views within this paper are those of the authors and do not reflect official policy of the author's DoD organizations or of the Government.

WHY OPEN SOURCE DEVELOPMENT?

“The Department (OSD) considers [information technology] IT critical to accomplishing its national security mission.”²

“A *New Approach for Delivering Information Technology Capabilities in the Department of Defense*” report to Congress was written in response to Section 804 of the National Defense Authorization Act for Fiscal Year 2010 (FY10 NDAA, PL 11-84) signifying the importance of IT acquisition reform. The report states:

*“In March 2009, the DSB (Defense Science Board) reported that the DoD was struggling to keep pace with the speed at which new IT capabilities are being introduced in today’s information age – and the speed at which potential adversaries can procure, adapt, and employ these same capabilities against the United States. The House Armed Services Committee (HASC) Defense Acquisition Reform Report of 23 March 2010 reached similar conclusions. ... The government is inefficient by design, burdened with a deliberate set of checks and balances, and clearly, it is time for the Department to review the sources of those inefficiencies and develop a new acquisition approach that is compatible with the fast-paced commercial IT sector and the evolving needs of the diverse DoD user base.”*³

Open source development methods have proven to be hugely successful in the “fast-paced” commercial IT sector. SourceForge.net, a commercial site that hosts an open source development environment, has

2.7 million developers who create powerful software in over 260,000 projects with more than 46 million consumers and more than 2,000,000 downloads a day.

⁴ This is only one of the sites where open source development is conducted. Well known, popular, and highly successful products such as Linux, Security Enhanced Linux (SELinux), Pegasus, Protégé, Lucene and Apache Web Services are developed and distributed using open source development methodologies. Open source software products are generally more reliable, secure, and user-based (customer focused). They are developed at significantly less cost, are more scalable than proprietary products, and produce groundbreaking complex technology at an accelerated pace.

OPEN SOURCE SOFTWARE - A MODIFIED APPROACH

Open Source Software (OSS) is generally characterized by a collaborative development method and environment with the following distribution criteria⁵:

- Free redistribution
- Program must include source code and must allow distribution in source code as well as compiled form
- License must allow derived works
- Integrity of the author’s source code
- No discrimination against persons or groups
- No discriminations against fields of endeavor
- The rights attached to the program must apply to all whom the program is redistributed
- License must not restrict other software
- License must be technology-neutral

Free redistribution, no discrimination, and no restrictions on software can be show-stoppers in DoD

² William J. Lynn III, Deputy Secretary of Defense, Opening Directive to “A New Approach for Delivering Information Technology Capabilities in the Department of Defense”, 9 December 2010

³ Office of the Secretary of Defense, “A New Approach for Delivering Information Technology Capabilities in the Department of Defense”, November 2010

⁴ <http://sourceforge.net/about>

⁵ http://en.wikipedia.org/wiki/Open-source_software

with current contracting, acquisition regulations, and security classifications. While there are material solutions developed by DoD that can comply to this wide open criteria, most cannot. That does not, however, mean that DoD cannot use the proven open and collaborative approach to software development and reap the same benefits. It does mean that a slightly modified approach grounded on the same open and collaborative principles that constitute OSS philosophies can be defined to provide the same benefits. The term Open Technology Development (OTD) has been coined for this purpose.

OTD APPROACH

OTD refers to the practices for development and implementation of technology that enables distributed groups of programmers to collaboratively develop and manage source development artifacts, such as architectures, designs, software code, and standards. OTD includes OSS and open source development methods, but also allows for limited distribution and appropriate considerations for the unique DoD development environment. It aims to answer the following questions⁶:

- How can DoD leverage military-funded software development more effectively?
- How can OTD's business-process advantages increase both the rate of innovation and the sustainability of software developed using DoD funds?
- What changes in acquisition practice and policy may be required to capture the benefit of OTD within and across the Defense Department?
- How can DoD leverage existing external OSS resources?

This paper will not answer these questions directly, but will use the OTD term and expanded concepts to refer to potential DoD implementation of open source development methods, give examples of previous and ongoing DoD OTD initiatives and provide some examples of innovative projects. It will propose approaches to begin changing the DoD culture and start capitalizing on the benefits of open source development.

⁶ J.C. Hertz, Mark Lucas, John Scott, "Open Technology Development", April 2006

IF OTD IS SO GREAT, WHAT'S THE HOLD UP?

There are several reasons open source development methods have not been enthusiastically adapted by DoD Program Managers, Project Managers, and leaders.

OTD and Internet Evolution

There appears to be a correlation between the internet evolution and open source development. While it is hard to find empirical evidence of this, it is easy to see many of the Web 1.0 and 2.0 technologies were developed through open source development methodologies. The internet has continuously reinvented itself through OTD practices, which foster innovation and development through communities of practice (COP). However, there has been no significant leap in DoD IT practices since the early 2000's when the DoD Net-Centric Data Strategy (NCDS) was created. In effect, DoD IT solutions are still at 1.0 while the internet is moving toward 3.0 and beyond.

The IT Government workforce is dominated by baby boomers and pre-baby boomers⁷ who are now in leadership positions. Their reluctance towards the internet evolution and OTD may be a contributing factor to the malaise state as depicted in Figure 1.

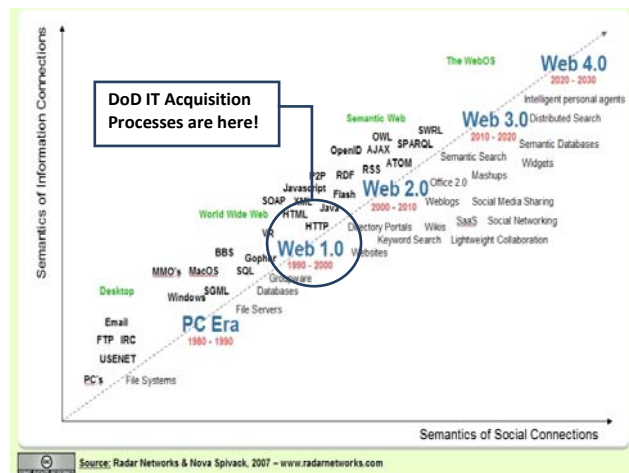


Figure 1 – Internet Evolution

Many of the new technologies and capabilities developed through the internet have been produced by the generation "x" and "y" digital natives. The "Net Gen", as they are called, is comfortable with OTD and

⁷ http://cio-nii.defense.gov/initiatives/netgenerationguide/current_environment.html

the fast-paced internet evolution, while the baby boomers and pre-baby boomers are not.

In order for the DoD to push past this state of malaise, and move toward the next generation, they must make use of their “Net Gen” talents, institutionalize OTD, and embrace internet technologies – at least until quantum computing makes its début.

Lack of Understanding

Most people, including DoD developers, find themselves humbly in awe when OTD makes order out of what appears to be chaos. They don’t understand how and why it works and definitely don’t see how it could work in a traditional sense. Eric Steven Raymond, a well known open software developer, wrote in reference to the Linux OSS development, “the Linux community seemed to resemble a great babbling bazaar of differing agendas and approaches (aptly symbolized by the Linux archive sites, who’d take submissions from anyone) out of which a coherent and stable system could seemingly emerge only by a succession of miracles.”⁸ In some organizations a campaign to educate and familiarize managers and leaders to the benefits and reasons why OTD works will be required to get buy-in and willing participation.

Contractual Intellectual Property (IP) Silos

Most current and prior DoD development contracts are not written to allow source sharing. Even when source is considered Government-owned, contracts are currently written in a manner that promotes sole source development. Most contracts don’t provision for modifications to be accomplished outside the scope of the contract. Future contracts will not only need to ensure language that allows for source sharing and open distribution, but also articulates tasking to focus more on configuration management, quality assurance testing and integration of source.

Licensing Heartburn

Although there are many different ways to license open source software and technologies, choosing the agreement that is most appropriate for a project is challenging and risky. Project Managers, with the help of DoD lawyers, must not only consider how to license the finished product, but take into account what constraints were agreed upon for accepting and re-using existing OSS. Fortunately, a lot of research has been done in this area, and many of the legal issues are

now better understood and resolvable. In addition, software analysis tools are being developed that analyzes, visualizes, and compares licenses to ensure that the OSS code does not have conflicting obligations prior to reuse and acceptance into a new architecture.

You Don’t Know, What You Don’t Know

Until recently, there was no mechanism for sharing source artifacts, such as code and project documents, across DoD organizations. Forge.mil is one mechanism that provides visibility and accessibility, but only if the development project is setup accordingly. An accessible, comprehensive repository of existing DoD IT capabilities is not available. Although there is a directive for all DoD organizations to establish and maintain an IT portfolio, there is no distribution policy, subsequently, these portfolios are neither openly discoverable nor searchable.

Security

Classification and releasability are often used as reasons not to use OTD. However, in most cases, the classified portion of a system is relatively small comparatively and is normally associated with the data and not the application. If systems are truly being designed as directed with open architecture, these sections of code ought to have the flexibility to be developed separately leaving the majority of the project available to use OTD methods in an unclassified environment.

While unclassified networks have tools that support OTD, most classified networks do not. “Need to know” and other security issues present hurdles in implementing classified OTD. Research still needs to be conducted to determine alternatives, and to assess the return on investment. Classified OTD may prove to be viable and valuable depending on the amount of classified code produced and its ability to be re-used.

Incentives

“Even if they could access source code across DoD, Program Managers currently have little or no incentive to do so. The current culture encourages and rewards based on budget execution and organizational size. In this environment saving time by re-using software would reduce their budgets (and thus possibly their prestige) and entail collaboration with a software community of practice, rather than status as sole master of their program domain.”⁹ However, current

⁸ <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>

⁹ J.C. Hertz, Mark Lucas, John Scott, “Open Technology Development”, April 2006

efficiency efforts and subsequent resource losses may propel the use of OTD to get programs completed. Acquisition lifecycle policy changes may also prompt the use of OTD for Program Managers to speed timelines and accomplish requirements.

Where To Get Started?

Many Program Managers don't know where to get started. Their organizations are not culturally aligned to accept OTD. For all the above reasons, organizational leaders often do not support individual project proposals to use OTD methods. Program Managers don't know what tools are available, are unsure what legal hurdles they may encounter, and are not sure they can assume the high risk of attempting a method they have not used before, especially when there is no precedence in the organization. The rest of this paper addresses this question – *How to get started?*

HOW TO GET STARTED

Awareness

The biggest and easiest problem to overcome is ignorance. Start by educating leadership on the value of OTD. Inform all members of the organization. Create a buzz.

While there might not be precedence in the immediate organization, there is precedence in DoD and other government agencies:

- National Security Agency (NSA) James Carter, Computer Systems Researcher, was directly involved with NSA's SELinux project. SELinux was NSA's initiative to get some of the more critical security enhancements into computer operating systems. Open Source (OS) provided a means to test and provide security enhancements for both the OS community (i.e., Linux) and proprietary OS developers. "The benefits we have seen are those of any OS project: our project is available to be used and studied (as a research organization, one of our goals), many external developers have contributed code and ideas, and we've been able to have far more impact than we would have otherwise."¹⁰

¹⁰ Weber, S., "The Success of Open Source", Cambridge, MA: Harvard University Press, 2004

- Delta3D, originally developed by the Naval Postgraduate School (NPS) is a completely open source project that is available via the internet.¹¹ It is a widely used and well-supported open source game and simulation engine. Delta3D is a full-featured game engine appropriate for a wide variety of uses including training, education, visualization, and entertainment. It is unique because it offers features specifically suited to the M&S and DoD communities, such as High Level Architecture (HLA), After Action Review (AAR), large-scale terrain support, and Sharable Content Object Reference Model (SCORM) Learning Management System (LMS) integration.

Vision

What is really expected to be gained from implementing an OTD environment? What is the scope of OTD? Will it apply to software only or all technology? Does it apply to materials only, or to collaboratively developing processes and standards? A complete understanding of what implementing OTD really means to the organization and the products and services produced by the organization will drive the vision. This is often difficult to articulate. Most statements are in terms of the benefits to DoD, the country, the taxpayer or the warfighter. However, an organizational focus is needed to internalize and make it real for the organization.

Plan

What approach will be used to create a functional OTD environment and employ an open technology standard in technology acquisition? How do you prepare the organization for a philosophical and cultural change? It will be easier to get leadership buy-in and sponsorship if a business or campaign plan is developed that clearly articulates the strategic and actionable approach to establishing, promoting, and maintaining an OTD culture. The plan should contain the following components:

- Leadership directive.
- The scope of participation in open technology development for the organization.
- OTD methods that may be employed.
- Technology acquisition and contracting guidance.

¹¹ www.delta3d.org

- Guidance on open source licensing that includes guidance on open source re-use and distribution within the organization and external to the organization.
- Considerations and guidance on security classification, information assurance, and information security.
- A deliberate communication plan.
- Defined strategies and specifications for standards; configuration management; quality assurance, test, and evaluation; and tools.

Building this structure will also ensure a thorough and well thought out plan for a more successful implementation.

Policy & Guidance

Establish policy and guidance across the organization, such that before any software or technology is resourced, an open technology development plan is required, to:

- Ensure market research is conducted as required by the Federal Acquisition Regulation (FAR)¹² and that open source alternatives are considered.
- Ensure a contracting strategy that reinforces the organizationally defined OTD environment and technology distribution.
- Ensure contracting strategy and language meets the OTD intent with proper legal review that ensures source re-use and public or government controlled distribution.

Develop a strategy to move pre-existing technology source into the OTD paradigm. This may mean re-competing contracts, establishing transition plans, establishing policy that mandates the retirement of proprietary technology at the end of its lifecycle, and other radical actions that reflect commitment to OTD. Of course sound decision making is required. There may be times OTD or open source technology is not the right choice, but this should be the exception rather than the rule.

Implement

There are several approaches to implement OTD. To simplify the options the differentiation between internal OTD and external OTD will be described. Internal OTD refers to an open technology

development environment for all technology developed through project teams inherent to the organization. Source is shared to all developers in the organization, but not released outside the organization without other arrangements. External OTD shares source with external developers that may include other DoD organizations, industry, academia, other government agencies, and the general public.

Organizations may start by creating an internal OTD environment. Allow the organization to “get-their-feet-wet” and test the waters. It should be noted that an internal OTD will not fully prepare for external OTD challenges, but it will provide the benefit of reduced costs through code re-use, community testing, and shared resources. This first step is a radical change for most organizations.

Tools/Resources

Several components are required for a successful DoD OTD environment:

- A common configuration management tool. This tool should provide the capability to manage source; define and articulate the purpose, function, standards, and architecture of the source; be searchable for developers to easily find source to re-use; be accessible from anywhere in the organization or where development is conducted; and collect metrics appropriate for performance evaluation of the OTD environment. Forge.mil is an example of this tool and can be found on the unclassified network. Unless security or classification prohibits, it is recommended that Forge.mil be considered as it allows for organizational sharing, but also provides access to externally developed OTD source.
- A tool for projects to analyze OSS licenses to ensure a conflict does not occur that prohibits the legal sharing and re-distribution of source.
- A legal resource to review contract statements and licenses to ensure the OTD environment does not incur arbitrary constraints.
- Templates, references, and examples of licenses, governance structures, standards, security requirements, project document templates, and other useful material that allows projects to stand-up quickly.
- Governance structures should be considered and shared with Program Managers. Will Project Managers be allowed to use a governance model that is tailored to the

¹² FAR part 10 – Market Research.

project, or will a specific model be mandated? There are several options of governance models ranging from the benevolent dictator to community vote. It is important to establish how project goals will be met, how source submissions will be accepted, and how conflicts will be resolved. For new organizations, a model that uses a core development team and benevolent dictator approach has the least risk.

Go For the Gusto

Whether the waters are tested first or the organization “jumps right in”, OTD will provide the most benefit with the largest body of water. There is benefit in numbers when it comes to technology development. The more developers and users of a technology the more refined, reliable, and secure it becomes. This has proven itself repeatedly in the public sector. Additionally, it greatly limits “vendor lock-in” by creating a large community of experts who will continue to maintain the technology as long as it is needed.

MOTIVATION

Motivation is the key to the successful implementation of an OTD. It is the difference between setting up an OTD environment and actually using it to share source and collaboratively develop technology.

Contracting Strategy

Current contracting strategy predominantly focuses on buying a product, not a service. In this case the government pays for a product and the contractor has some level of intellectual property rights and the legal obligation to produce a product that meets requirements. Besides limiting the re-use of technology source materials, like code, a lot of risk is put on the contractor, making the focus of the effort to get done just enough to meet timelines. The fewer resources or less effort they expend building the product, the more profit they get. There is little or no motivation to be innovative or to go beyond expectations. No wonder the public sector technology is way ahead.

To claim OTD’s benefits of rapidly produced innovative solutions, a new paradigm is required. *Services vice products* should be contracted to develop and maintain project standards and architecture; establish and perform configuration management control; develop and maintain project and product

requirements; develop criteria for source acceptance; perform the function of source test, evaluation, and acceptance into the baseline; ensure information is available and open to the community; and baseline product development. By contracting the service and not the product, the contractor is free to be innovative within the guidelines of the program and recognizes that they do not have sole intellectual ownership of the product. Compensation is relative to how well services are being performed to facilitate the successful development of the product. Monetary incentives may even be given for key project successes. This approach does add risk to the government even with the careful consideration and development of performance criteria. The Program or Project Manager assumes more responsibility for the successful production of the product.

Program and Project Managers

So, what motivates program or Project Managers to want to employ OTD methodologies when it means assuming more risk? The first and foremost motivation is the prospect of bringing better capability, faster and cheaper to the warfighter and helping to ensure their success in defending our freedoms. IT is critical to our national security.¹³ OTD has been identified as a critical catalyst for the DoD to bring innovation to our warfighter. The recognition alone for managing a program or project that brings rapid innovative ideas to the warfighter will not go unnoticed. Additionally, because OTD is still somewhat novel to the DoD, areas for improvement provide great opportunities to Program and Project Managers to publish their lessons learned. Managers will also get recognition for producing source materials that can truly be re-used and re-distributed without intellectual property headaches.

Development Community

Critical to the equation of a successful OTD is the development community. Developers can be one-way participants or two-way participants. They can re-use code openly distributed in new projects, or they can re-use source and then contribute back beneficial enhancements. One-way developers are easier to understand. They can develop their product faster and at less cost by using existing source and building on it. Two-way developers are harder to understand. What is the motivation for contributing back to a project the

¹³ William J. Lynn III, Deputy Secretary of Defense, Opening Directive to “A New Approach for Delivering Information Technology Capabilities in the Department of Defense”, 9 December 2010

developer has nothing to do with other than using the source product given away freely?

There is a moderate amount of research material attempting to understand the motivation of open source developers in the general public setting. Dan Pink says that when people are paid enough, such that it is not a worry to them, they are motivated by the “desire to be self directed” (autonomy), the “urge to get better at stuff” (mastery), and “some kind of transcendent purpose” (purpose)¹⁴. The Department of Defense work has the inherent transcendent purpose of supporting the warfighter to defend and protect our country and the freedoms afforded by our constitution. For this reason many independent developers will gladly contribute, considering it for a good cause.

Independent developers that develop a reputation for providing good source are often afforded top job opportunities. The bigger the project they contribute to and the more recognized the project, the more visibility to contributing developers. For example, being known as a star developer of the much publicized OSS Linux operating system has some weight. Of course this means that a project will attract a larger development community if it is more known. Project Managers will want to consider this when deploying source for OTD in the public environment and plan to advertise the project on key development sites and in internet forums.

Most DoD OTD development will not be done in a totally open, public forum. Most will likely be limited to the .mil domain or an organizational domain. Generally, technology is developed by contractors in these domains. How do you motivate a contractor to want to produce source and then give it away with no intellectual property rights? One way is to establish a monetary award for accepted contributions to a project. The contributing information would be annotated for recognition and required to stay with the source. When a contract is required, one of the technical evaluation criteria should be based on the re-use of existing open source as well as the intention to make developed source open and re-useable. If it is not in the contract, it won't happen.

OTD SUCCESS STORIES

Ongoing DoD R&D projects described in the following sections illustrate innovative strategies, OTD practices, and their potential to alleviate many legacy system interoperability issues for transitioning to the next generation DoD IT infrastructure.

¹⁴ <http://www.youtube.com/watch?v=u6XAPnuFjJc>

Coalition Battle Management Services (CBMS)

CBMS is a technical infrastructure that enables the exchange of resources between Command and Control (C2) and M&S systems, and robotic forces. Initial use cases include: exchange of orders, reports, and requests between fielded legacy C2 and M&S systems; After Action Review (AAR) support and visualization capability to support a Common Operational Picture (COP); data distribution management; persistent store (XML data store) with respective metadata to provide resend/replay capability; time management to track and synchronize message passing for improved situational awareness; and parametric search/filtering to locate and provide relevant-only information.

CBMS uses an open architectures/OSS design philosophy. It will be accessible via any commercially available web browser and uses only next generation XML-based technologies in its implementation as depicted in Figure 2. It is system-independent, allowing each consumer or producer system to map their respective system language to another language.

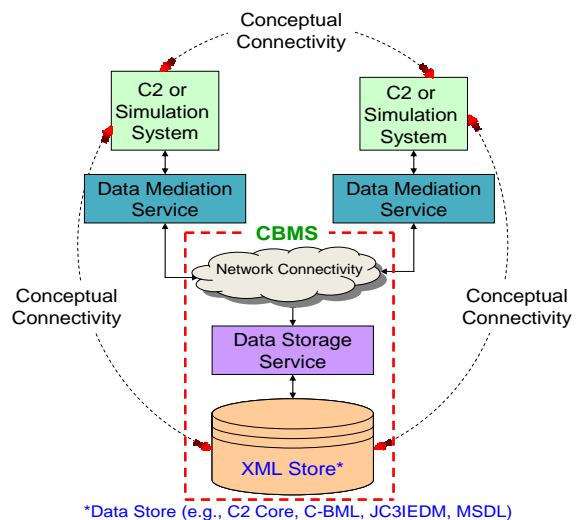


Figure 2 – CBMS Diagram

The Simulation Interoperability Standards Organization standards committee is currently reviewing the CBMS enterprise architecture. CBMS leverages only open source web technologies:

- Xbase for document persistence and XQuery processing.
- Atmosphere framework for HTTP-based messaging.
- Jersey for RESTful web servicing.
- xLightweb for client-side HTTP processing.

In contrast to the kluge of disparate architectures, bridges, gateways, and data sharing strategies currently used by the DoD M&S community, CBMS is a decoupled collection of composable web services that can be orchestrated to support the needs of a particular federation. This design facilitates rapid technology refresh and encourages reuse.

Currently, the CBMS suite of tools is being further developed and refined to accommodate an OTD environment with Coalition partners. Executable code and documentation have been provided to Coalition partners through The Technical Coalition Program and the NATO Modelling and Simulation Group to facilitate peer review and product feedback.

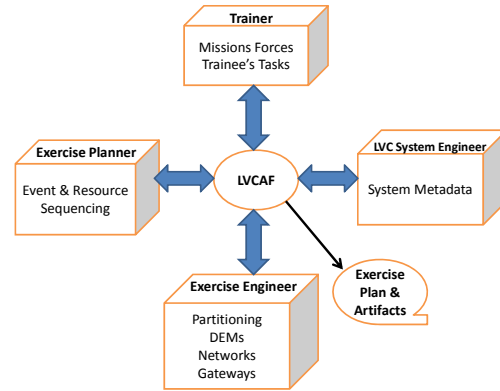


Figure 3 – LVCAF Diagram

Live, Virtual, Constructive Framework (LVCAF)

LVCAF is a framework that supports search, discovery, and composition of federation components from multiple architectures while providing linkages to functional mission capabilities. It uses ontologies as a common vocabulary to facilitate machine-to-machine communication and a knowledge base to simplify reconciling models and promote reuse. LVCAF translates object models between the following disparate DoD M&S architectures:

- High Level Architecture (HLA) Federation (1.3, 1516 and 1516 evolved)
- Test and Training Enabling Architecture (TENA) Logical Range Object Model (LROM)
- Distributed Interactive Simulation (DIS) Protocol Data Units
- Common Training Instrumentation Architecture (CTIA) object models

LVCAF stores semantically matched components in composed data exchange models (DEM) with linkages to mission threads to facilitate event execution as shown in Figure 3.

LVCAF utilizes standard Semantic Web formats:

- eXtensible Markup Language (XML)
- Resource Description Framework (RDF)
- Web Ontology Language (OWL)

LVCAF leverages only open source web tools and development techniques:

- Archiva for repository management.
- OntoWiki for project collaboration.
- Mercurial for code versioning.
- Protégé for the inference engine and knowledge representation.
- JIRA w/ Greenhopper for project management, managing scrum backlog, planning sprints, and release tracking.
- Agile software development methodology with 30-day sprints and daily scrums.

LVCAF provides an object model comparison capability to reduce manual resource intensive object model reconciliation by the DoD M&S SME.

LVCAF is finalizing the OTD process to launch the tool suite for public development and consumption.

Organizational Open Development (Enterprise Source Software)

The Aerospace Corporation is a federally funded research and development center for national security, civil, and commercial space programs. In an effort to bring modern software development practices and discipline to their more “ad-hoc development projects”, they incorporated open source development

methodologies into the organization. They call this internal open source development “Enterprise Source Software” (ESS). Source code is free to anyone in the enterprise to read, build, run, make changes to it, and redistribute within the company. “External software distribution remains governed by existing processes.”¹⁵

They deliberately promoted the open source philosophies, development methods, and tools through Education. Education included meetings, lectures, and tutorials. They provided an extensive development environment built on open source tools targeted at open source development methods. Their grassroots promotion has led to many projects acceptance and full use of the ESS environment.

SUMMARY

The idea of open source development is the farthest thing from most DoD Project Manager’s minds when it comes to developing material solutions. It is too risky, too unfamiliar, too many security concerns, too many licensing issues, and just too many questions. Nevertheless, the DoD cannot continue to be crippled by excuses. They cannot continue to be “inefficient by design”. They cannot continue to struggle “to keep pace with the speed at which new IT capabilities are being introduced in today’s information age – and the speed at which potential adversaries can procure, adapt, and employ these same capabilities against the United States.”¹⁶ Rapid technology growth is directly linked to the open source development methods widely used on the open internet. This paper was written to alleviate some of the trepidations withstanding and provide doorways for the DoD to embrace open source development and capitalize on the resulting rapidly produced, reliable, secure, and user-based technology.

The paper discussed the primary arguments heard by Project Managers for not using open source development methods, discussed how DoD can use open source development methods in the form of Open Technology Development (OTD), and provided a roadmap with examples to get started. Program Managers, Project Managers, and developers have a launching point to begin planning their IT development revolution. To start the wave and excitement of progress in their organization and across the DoD. To outsmart the potential adversaries!

¹⁵ Davis, Brooks, Michael AuYeung, and Mark Thomas, “Using FreeBSD to Promote Open Source Development Methods”, The Aerospace Corporation, 2007-2008

¹⁶ Office of the Secretary of Defense, “A New Approach for Delivering Information Technology Capabilities in the Department of Defense”, November 2010

The views within this paper are those of the authors and do not reflect official policy of the author's DoD organizations or of the Government.

ACKNOWLEDGEMENTS

The authors would like to acknowledge Mr. Allen Lawver and Mrs. Denise Stenstrum of Joint and Coalition Warfighting for their review and encouragement. Mr. Mark Phillips (Lockheed Martin, Inc.), Dr. Barry Smith (University at Buffalo), Mr. Scott Streit (Intervise, Inc), and Dr. Lowell Vizenor (Alion Science and Technology) for their invaluable input through their work on the *Independent Study of the Semantic Web, Ontologies, and Initiatives for Realizing the Next Generation Defense Training Environment* published 21 February 2011 sponsored by the United States Joint Forces Command. In addition, we would like to thank Mr. Ed Kulakowski from OT Training Solutions, Inc. for his great advice and advocacy.

REFERENCES

- Chief Information Officers Council, (2010). *Net Generation*. Retrieved from http://cio-nii.defense.gov/initiatives/netgenerationguide/current_environment.html.
- Davis, Brooks, Michael AuYeung, and Mark Thomas, (2007-2008). *Using FreeBSD to Promote Open Source Development Methods.*, El Segundo, CA: The Aerospace Corporation.
- Delta3d.org, (2011). *About*. Retrieved from www.delta3d.org.
- Hertz, J.C., Mark Lucas, John Scott, “Open Technology Development”, April 2006.
- Jacobs, I, Description of W3C Technology Stack, February 03, 2010, Retrieved from: <http://www.w3.org/consortium/techstack-desc.html>.
- Joshi, Bela, Ph.D. and Curtiss Murphy, (2007). *Are You Ready? The Open Technology Development Challenge*. IITSEC Paper No. 7100.
- Lynn, William J. III, Deputy Secretary of Defense, (2010). *Opening Directive to A New Approach for Delivering Information Technology Capabilities in the Department of Defense*.
- Nitrd.gov, (2011). *About*. Retrieved from http://www.nitrd.gov/About/about_nitrd.aspx.
- Office of the Secretary of Defense, (2010). *A New Approach for Delivering Information Technology Capabilities in the Department of Defense*.
- Pink, Dan, (2010). *What Motivates People to Contribute to Open Source?*, (David M. Turner).

- Retrieved from, <http://www.youtube.com/watch?v=u6XAPnuFjJc>
- Raymond, Eric Steven, (2000). *The Cathedral and the Bazaar*, Retrieved from <http://www.catb.org/~esr/writings/cathedral-bazaar/cathedral-bazaar/index.html>.
- Scacchi, Dr. Walt, Thomas A. Alspaugh, Hazel Asuncion, (2010). *Investigating the Acquisition of Software Systems that Rely on Open Architecture and Open Source Software*, (Acquisition Research Sponsored Report Series UCI-AM-10-021). Irvine, California: Institute for Software Research, University of California, Irvine (U.S. Government or Federal Rights License).
- Scott, John, David A. Wheeler, Mark Lucas, J.C. Herz, (2011). *Open Technology Development (OTD): Lessons Learned and Best Practices for Military Software*. Sponsored by the Assistant Secretary of Defense (Networks & Information Integration) (NII)/DoD Chief Information Officer (CIO) and the Under Secretary of Defense for Acquisition, Technology, and Logistics (AT&L).
- SourceForge.net, (2011). *About*. Retrieved from <http://sourceforge.net/about>.
- Stenbit, J. P., (2003). Department of Defense Memorandum, *DoD Net-Centric Data Strategy*.
- Turner, David, (2010). *What Motivates People to Contribute to Open Source?*. Retrieved from, <http://davidmturner.com/blog/what-motivates-people>.
- Walsh, Damien and John Masterson, (2007). *What is the Ruckus About DoD Issuing Open Source Software?*, CHIPS magazine, Volume XXV Issue IV, pg. 44 – 45.
- Wheeler, David A., (2010). *Open Source Software (OSS) in U.S. Government Acquisitions*. Retrieved from <http://www.dwheeler.com/essays/oss-government-acquisitions.html>.
- Wikipedia, (2010). *Ontology (information science)*. Retrieved from [http://en.wikipedia.org/wiki/Ontology_\(informaton_science\)](http://en.wikipedia.org/wiki/Ontology_(informaton_science)).
- Wikipedia, (2011). *Open Source Software*. Retrieved from http://en.wikipedia.org/wiki/Open-source_software