

report



Open Source Software in Defense Markets

Issues Surrounding OSS Usage in C4ISR Roles

May 30, 2008

Corporate Services Practice
Tate Nurkin
110 N. Royal Street, Suite 200
Alexandria VA, 22314
+ 1 (703) 236 2417

Table of Contents

TGlossary of Acronyms	1
Introduction and Executive Summary:	2
The Open Source Debate	4
<i>Convergence and a Hybrid Approach</i>	9
The Open Source Debate in the Military Context	11
<i>Trends Driving Software Procurement and Use in the Military</i>	11
<i>Presence of and Emerging Attitudes Toward OSS in Government and Military Environments</i>	12
<i>Assessment of Software Solutions in the Emerging Military Context</i>	14
<i>The Technology Hierarchy and Emerging C4I Requirements</i>	21
Conclusion	23
Appendix A: Trends Driving Software Procurement & Use in Military	24
Appendix B: Network-Centric Characteristics and Definitions	25
Appendix C: Origins of Open Source Software	26
Appendix D: LP Committee License Category Definitions	28
Appendix E: Individual LP Committee Licenses by Category	29
Appendix F: Number of UNIX/Linux C4ISR Platforms by Country	31
Appendix G: Approved National Government OSS Policies	32
Appendix H: Taxonomy of Costs	1
Appendix I: About Network Centric Warfare	44

Glossary of Acronyms

Acronym/Abbreviation	Definition
AI	Artificial Intelligence
API	Application Programming Interface
ASD [NII]	Assistant Secretary of Defense for Networks and Information Integration
BSD	Berkley Software Distribution
C++	A Computer Programming Language
C4I	Command, Control, Communication, Computers, and Intelligence
C4ISR	Command, Control, Communication, Computers, Intelligence, Surveillance, and Reconnaissance
CIO	Chief Information Officer
CONOPs	Concept of Operations
COTS	Commercial Off-the-Shelf
CS	Commercial Software
DARPA	Defense Advanced Research Projects Agency
DISA	Defense Information Systems Agency
DKO	Defense Knowledge Online
DoD	US Department of Defense
GIG	Global Information Grid
GNU	Gnu's Not Unix
GPL	General Public License
HUMINT	Human Intelligence
IA	Information Assurance
IP	Internet Protocol
ISR	Intelligence, Surveillance, and Reconnaissance
IT	Information Technology
MIT	Massachusetts Institute of Technology
NAO	UK National Audit Office
NCES	Net-Centric Enterprise Services
NCW	Network Centric Warfare
NECC	Net-Enabled Command Capability
OSI	Open Source Initiative
OSS	Open Source software
RDBNS	Relational Dynamic Bayesian Networks
RFP	Request for Proposals
TCO	Total Cost of Ownership
XML	Extensible Markup Language

Introduction and Executive Summary:

The debate between OSS and commercial software providers has been the most central debate in the software and IT industry over the past half-decade. In March of 2008, Microsoft commissioned Jane's Strategic Advisory Services (JSAS) to author a paper designed to address three key issues related to how arguments associated with the OSS vs. commercial software debate might translate to the military context: first, to outline the current and historic contours of the general debate between OSS and Commercial Software (CS) providers, particularly in the commercial environment; second, to examine attitudes towards OSS and prevailing trends in software procurement in the emerging military context, including identifying unique priorities for evaluating software in a military context; and finally, to examine how future C4I requirements might drive procurement of OSS or CS solutions at varying levels of the technological hierarchy.

While this paper was commissioned by Microsoft, the goal of this paper is *not* to advocate for or lobby against a particular approach or provider. As a division of Jane's Information Group, an organization with a long-standing and valued reputation for independent research, analysis, and reporting, JSAS is acutely aware of and dedicated to the principle of maintaining analytical objectivity and impartiality, particularly on commissioned projects. This paper's main goal is to provide the research team's best assessment of the nature of the debate between OSS and CS providers and of the issues and factors that distinguish software procurement and use in the military from the commercial context. Jane's research for this paper included an extensive survey of secondary source materials as well as conference attendance and primary source interviews with senior US Department of Defense personnel and advisors who are deeply familiar with DoD approach to OSS; defense industry experts; technologists in the United Kingdom, the United States and Israel; and advocates of both OSS and CS approaches. Below are a series of key insights that emerged over the course of Jane's research.

Convergence of Models: A key trend in the current software and IT environment is the convergence of OSS and CS approaches. While significant differences do and will continue to exist, both open source and commercial providers are increasingly taking hybrid approaches to providing software and software support. Some open source providers, such as Red Hat for example, have sought to monetize OSS in ways that are not always compatible with long-held open source movement principles and have also created support teams that are in many ways not dissimilar to the CS model. Commercial providers have taken various approaches to "opening up" their software and are either adopting OSS solutions or increasingly allowing users to examine and monitor code and provide feedback about this code to support teams directly through established forums.

OSS in the Public Sector and the Military Context: Public sector use of OSS has increased since 2002, with several governments seeking to encourage or mandate municipal use of OSS through incentives or legislation. However, enthusiasm for OSS in the public sector has waned since 2006, and states and militaries are examining the trade-offs between the possibility of a lower *up-front* acquisition cost of OSS (although not the overall TCO¹) and the perceived enhanced support provided by CS providers. Thus, while militaries around the world use OSS or Unix-based operating systems, attitudes about OSS in the military context have yet to fully crystallize and range from notional acceptance and ad hoc use to uncertainty and circumspection

¹ TCO – Total Cost of Ownership, other costs such as support, maintenance, development & integration

to rejection and resistance. A mix of all three pervades many services and departments and ministries of defense.

Evaluation Criteria in the Military Context: Software procurement decisions in the emerging military and

operating environments are driven by a set of distinct, often demand-side, considerations, such as those listed in the text box to the right.

Analysis of recent software procurement

- The nature of the military requirement that the software is being developed to meet
- The timing of the deployment of platforms or forces and the nature of their roles
- The ability of the provider to support the software over its lifetime
- The security of the software
- The ability to adhere to military classification and security standards
- The ability of the software to facilitate interoperability of joint or coalition forces
- The ease of use and management and resilience of the software over the life of the program, particularly in restricted, closed and classified communities
- The ability of armed services and national security organizations to recruit and retain large enough and capable enough development communities
- The level of risk aversion associated with the end-user
- Procurement history and budgetary or political restraints of users and total cost of ownership

decisions in several states that are aggressively transforming to a more network-centric force structure suggests that, after security, militaries seem to be most concerned with the supportability of the software, the pedigree of the provider, the ability of the software to adapt to new and evolving operational requirements, the total cost of ownership, the ease of use and deployment of the software, and the systems and platforms the software supports.

A Dead Heat: Security and Software: While security is the primary consideration in military environments, neither OSS nor CS is more or less secure in every instance. Consideration of security, then, must be made on a case-by-case basis with a fundamental understanding of the specific security and mission requirements of the platform or system that is being developed.

Emerging C4I Requirements and the Technological Hierarchy: For those states most aggressively seeking to transform to a more network centric force, a key priority will be developing technologies that will allow for the conversion of the abundance of information currently available into relevant, easily accessible, and sortable knowledge for operators. OSS may have a role to play in this process, however, Jane's research reveals that CS or hybrid solutions appear better suited to drive the innovation at the higher levels of the technology hierarchy—the meta-data layer, net services layer and applications layer—that provides for this conversion from information to knowledge.

The Open Source Debate

The emergence and efficacy of ‘open source’ concepts in the software and computing fields over the last decade-plus is one of the most important and impactful trends within the software industry. This has been particularly true in the commercial software industry where the emergence of Open Source Software (OSS) based-systems has posed an alternative to traditional providers of Commercial Software (CS).

Because the open source software movement was initially established as a means of providing alternatives for users of traditional CS providers—initially IBM and now primarily Microsoft, for users of operating systems—the OSS debate is most often perceived as a zero-sum game competition between OSS systems and commercial software providers. Thus, as OSS has become increasingly prevalent in certain market sectors, some commercial enterprises have come to consider the open source movement and the success it has achieved as a challenge to their business models. Other companies, alternatively, have embraced OSS-based systems and products.² It is this debate about OSS and, more specifically, the relevance of OSS in the emerging military context—both as operating systems and applications—that the following paper will explore and examine.

Before establishing the framework of the OSS debate, it is essential to define the most important terms in this debate: OSS and CS. Both concepts are tied to the source code of a program. Open Source Software (OSS), in its most basic terms, implies that the original source code of a program is openly and freely available to the public to be viewed, modified and redistributed by its users within 10 clearly identified redistribution criteria.³ CS, on the other hand, implies that the original source code of a program was created by certified, commercial developers who subsequently develop software programs that are resold.⁴ OSS is open in the sense that users can see and, in many cases, modify the code to meet their requirements, as long

The 10 Criteria for Open Source Compliance

1. **Free Redistribution:** Licenses shall not restrict any party from selling or giving away software as a component of an aggregate software distribution containing programs from several different sources
2. **Source Code:** Program must include source code, and must allow distribution in source code as well as compiled form
3. **Derived Works:** License must allow modification and derived works, and must allow free redistribution of these works
4. **Integrity of Author’s Source Code:** License may restrict source code from being distributed in modified form only if the license allows the distribution of patch files with the source code for purposes of modifying the program at build time
5. **No Discrimination against Persons or Groups**
6. **No Discrimination against Fields of Endeavor**
7. **Distribution of License:** The rights attached to the program must apply to all to whom the program is redistributed without the need for execution of an additional license by those parties
8. **License Must Not Be Specific To A Product**
9. **License Must Not Restrict Other Software**
10. **License Must Be Technology Neutral:** No provision of the license may be predicated on any individual technology or style of interface

² Darius Hedgebeth, *Gaining competitive advantage in a knowledge-based economy through the utilization of open source software*, Emerald Group Publishing Limited; Vol.37, Iss.3, 2007

³ The Open Source Initiative, <http://www.opensource.org/docs/osd>

⁴ The definitions for OSS and CS comply with the Microsoft Corporation definitions, “Perspectives on Microsoft and Linux,” Microsoft Corporation, 2006.

as the source code is included in any modifications that are distributed. CS traditionally does not allow users to see and / or modify the code that is being sold. All modifications are done by the provider. In addition to these two approaches—OSS and traditional CS—some commercial providers have made significant alterations to the traditional CS approach, which allows users to see and provide feedback on the source code through established forums. The development of this hybrid approach by Microsoft is addressed thoroughly later in this section

Key issues in the debate between OSS and CS providers have focused on the relative opportunities, challenges and risks associated with using OSS or CS. Below is a brief overview of these key issues associated with the broad debate between OSS and CS. These issues and how they relate to software use in the emerging military environment will be addressed in far more detail later in the paper.

Openness and Development: Just as open science makes for good science, OSS is rooted in the principle that open programming will make good programming.⁵ The openness of the code and development process, according to those who support the OSS movement, allows for enhanced security and for more efficient and cost-effective code, since all users can assist in identifying and detecting bugs and inefficiencies. This development process relies on both the size of the development community and how well organized (or not) it is. CS providers have noted that keeping a user community together in the OSS context is difficult, as is verification and validation of who is involved in the development and testing process. Additionally, the more specialized the software being written is, the less likely a developer will be able find and maintain a usable development community.

Diversity and Modification: OSS proponents argue that open source software promotes diversity and competition in the market place and that this diversity and competition can only provide increased security, reliability, efficiency and lower costs to the end-user. OSS proponents also point to the core principle that OSS can be modified by users as they see fit as a particular strength that allows OSS to be tailored directly by the users to meet specific user-requirements. CS providers counter this argument by pointing out that this modification also may require users to support this modified version.

Lock – In: Since OSS source-code is freely available, advocates point out that there is no danger of ‘lock-in’ by a CS provider. CS providers counter that the chance of “lock-in” is not as great as OSS advocates believe due to the “quite amazing”⁶ pace at which the current cycle of technology is being outdated. CS providers also state that the real chance of lock-in is reduced especially if the system used advocates open standards, as it becomes considerably easier to “surround the technology.”

Security: Both sides assert that their approach to security is more effective. OSS proponents suggest that the openness of the code allows for enhanced scrutiny and, therefore, for enhanced detection. Security, of course, is not solely about preventing and detecting bugs. It is also about defeating them once they emerge. And in this regard, OSS proponents again argue that the openness of their process allows for quick and efficient releases of patches and fixes. Conversely, CS providers believe that openness may create security risks by

⁵ *Ibid*

⁶ Email Interview with Microsoft personnel, June 3, 2008

allowing those intending to release exploits to see both the plausible strengths and vulnerabilities of the code they wish to attack. Thus, keeping a closer hold on the code helps mitigate this risk. In addition, CS providers also argue that while the OSS mantra is to “release early and release often” the CS approach does a highly effective job of fixing vulnerabilities and releases patches at a high frequency as well.

CS providers also make the argument that while a large community may lead to increased security scrutiny of OSS in theory, there may be gaps that emerge as developers focus on developing new functionalities rather than on the more mundane work of detecting bugs. As John Viega, co-author of Secure Programming Cookbook for C and C++ wrote: “Most people who look at the source code for open source software don’t explicitly look for security bugs. Instead they likely have in mind a particular piece of functionality that they want to augment, and they’ll look at a subset of the code to scope it out and maybe write a fix. Generally this does not require a thorough audit of the entire program. This kind of person might casually notice a security bug in the software, but it’s probably best to assume it’s a long shot.”⁷

This statement was written in 2004 and while several open source providers have developed security reviews for code since this time, the potential problems associated with open source review have been starkly seen and felt in the recent past. Most notably, in September of 2006, a flaw was introduced into the Debian OpenSSL code, which, when combined with the cascading interdependencies of open source code, led to massive vulnerabilities that went undetected until May of 2008. The flaw in the random key generation function was introduced by one code reviewer who detected a flaw using an automated tool. The adjustment of this code, however, led to a dramatic reduction in the entropy of the security keys produced, and because the code was changed only slightly, this defect was not detected for nearly 20 months.⁸ This flaw means that OpenSSL keys generated from September 2006 until May 2008 were “so predictable that an attacker can correctly guess them in a matter of hours.”⁹ Some estimates of the number of keys required to be replaced were as high as “hundreds of thousands or millions.”¹⁰

The Debian debacle, of course, represents a near (though not complete) worst case scenario of the vulnerabilities associated with the open source approach to security and has little comment on the vulnerabilities in CS solutions. However, for proponents of CS solutions, it does provide a reminder of the flaws associated with Linus’s Law that “more eyes make all bugs shallow.”

Total Cost of Ownership (TCO): Evaluation of TCO of CS and OSS involves several layers of cost analysis. In addition to the ‘up-front’ hardware and software acquisition costs, a true cost-benefit analysis must take into account additional factors, such as *staffing, training, downtime, outsourced functions, support, and ongoing management*. Therefore, while open source software can provide considerably lower initial acquisition costs and cheaper per-seat scalability, the cost of migration from either option to the other can be

⁷ Viega, John, *Open Source Security: Still a Myth*, O’Reilly on-line publishing (www.oreilly.com), www.onlamp.com/pub/a/security/2004/09/16/open_source_security_myths.html, 9/16/2004

⁸ Information gathered from multiple sources, including: Goodin, Dan, *After Debian’s Epic SSL Blunder, A World of Hurt for Security Pros*, The Register, www.theregister.co.uk/2008/05/21/massive_debian_openssl_hangover/, 21 May 2008. and Appelbaum, Jacob, *Crippling Crypto: The Debian OpenSSL Debacle*, presented at the Hackers on Planet Earth (H.O.P.E) Conference held in New York City in July 2008

⁹ Goodin, Dan, 21 May 2008

¹⁰ *Ibid*

extremely high. So, while the initial cost of procuring CS systems in most cases is higher than OSS, the low procurement cost of OSS is ultimately only a fraction of the cost that OSS users are likely to incur. Administrative and support costs such as those incurred transitioning from a CS system, training on difficult to navigate interfaces, obtaining software support, through downtime of systems as patches are released, integrating OSS and high-end CS applications, and managing the software, all contribute to the cost of an OSS procurement. However, some of these costs are not paid explicitly in terms of money, but rather in terms of time.

Whether open source software is less costly to administer than traditional CS depends largely on the nature and size of the pool of resources trained on the system, the availability of administration tools that allow system administrators to manage a greater number of systems, and the number and effectiveness of version upgrades and patches that are issued by the developer. In this regard, the advantages in costs accrued by open source software that is less expensive to procure are diminished, although this calculation will vary from application to application.¹¹ In addition, when considering cost it is also important to consider the amount of time one expects an operating system to be in service, costs that will be recurring over the life of the system, non-monetary costs, and those that will be paid once or that diminish over time.

Supportability and Reliability: Among the most prominent arguments made by CS providers is that CS systems provide a considerably higher degree of support than OSS systems, which rely on a broad and disparate community for support. Indeed, the community nature of OSS development makes software support, in a traditional sense, less centralized in many cases (Red Hat and others have taken a different approach to support, discussed below) because an individual user does not have a central authority to turn to when problems arise. For example, online community forums have become a major avenue used by the OSS community at large, which are occasionally cheaper and faster than calling on CS providers. Support issues can find resolution through community interaction on such forums, but these forums do bring up questions of the reliability of the community posting on the forum. That is, what validation does a user have that individuals on forums are knowledgeable or well-intentioned? OSS proponents argue that these same questions of identity, intention, and acumen could be applied to CS provider support center personnel as well. However, as CS providers note, these individuals do go through varying degrees of vetting and are responsible to a central body, the CS provider. In addition, such forums would be difficult if not impossible to access and use in the classified or secure environments in which military operators most often operate.

Some OSS-based businesses, such as Red Hat, SUSE, or Ubuntu, have adopted an approach that is closer to the support approach of CS-providers by employing teams of programmers to provide dedicated support. This support is sold as a service, which adds to the total cost of ownership of the software, but also helps the commercial provider capture increased value. The Open Source Think Tank Conference of 2007 estimates that the top 20-30 commercial open source vendors offer SLA-level support and third party vendors such as OpenLogic “are stepping into the void” by offering SLA support and indemnification for many other open source components.¹² However, at this event, it was pointed out that other open source applications,

¹¹ <http://www.computereconomics.com/article.cfm?id=1043>

¹² 2007 Open Source Think Tank: The Future of Commercial Open Source Executive Summary Report, March 8 – 10, 2008.

components, and libraries that “sit above the operating system layer and below the application layer lack commercial support options,” which is “something the open source industry will have to address.”¹³ One potential issue with this approach is that this approach can blur the line between OSS and commercial vendors and plausibly create an environment in which companies could be using multiple versions of Linux or other OSS that are being exclusively supported by different vendors.

Usability: Another argument made by CS providers, particularly Microsoft, is that CS user-interfaces are considerably easier to navigate than those in OSS, and that most users are far more familiar with traditional CS interfaces. The more complicated and unfamiliar OSS graphical interfaces actually could increase the TCO of OSS systems by enhancing downtime and training requirements to bring users up to speed on OSS systems.

Flexibility: OSS advocates point to flexibility in procurement and deployment as a key advantage of OSS in commercial environments as companies can acquire OSS from a variety of sources and can mix and match open source software far easier than when dealing exclusively with CS providers.¹⁴ Many CS providers argue, however, that purchasing different OSS at different levels of the technology hierarchy from different providers with different support models can create challenges and confusion in terms of support.

Licensing: Licensing has long been a somewhat contentious subject for both the OSS and CS community. As referenced above, the OSS community has argued with great vigor that CS providers “lock-in” users with long and restrictive licenses that inhibit user autonomy. Licensing has also been a tricky issue for OSS, though. As awareness and interest in the open source concept developed and approaches to open source software evolved, the proliferation of open source software licenses created confusion over the nature of OSS licensing. To alleviate confusion about various types of open source licenses, the License Proliferation (LP) Committee of the Open Source Initiative grouped all licenses into six categories, which are explained in further detail within Appendix D. Appendix E lists the individual licenses included in each category defined by the LP Committee.

Even within the LP Committee categories, various approaches have been employed that blur the line even further between license types. The various licensing types used in open source software projects range from a very flexible BSD (Berkley Software Distribution) type license that has no obligations other than publication of copyright notice and sometimes only requires acknowledgment of open source use, to by far the most widely held license, GPL (GNU General Public License), that requires source code for modifications that an individual or business distributes to be made available under GPL.¹⁵ A number of commercial firms have adopted a variety of methods to license their products in an open source manner. In short, an increasingly commonly held view even within the open source community is that “open source licensing is a big source of confusion due to the number of open source licenses and a lack of understanding of how licenses impact business as well as how licenses interact with one another.”¹⁶

¹³ *Ibid*

¹⁴ *Ibid*

¹⁵ Douglas Heintzman

¹⁶ 2007 Open Source Think Tank: The Future of Commercial Open Source Executive Summary Report

The adoption of OSS by for-profit companies, however, appears to have become a source of contention between the OSS community and commercial organizations that are attempting to monetize open source software. In fact, the Software Freedom Law Center recently launched four lawsuits against for-profit organizations (the largest of which is Verizon) for violation of the most recently released version of the GPL license, GPL3.¹⁷ One intellectual property commentator notes that these suits are not necessarily about changed provisions found in GPL3, but rather they are reflective of changes in the way that commercial entities are using OSS, namely leveraging software for business advantage, which is, in turn, likely to restrict the freedom of software that is intended to be free. As a June 3, 2008 Law.com article notes:

“Implementing proprietary features on top of open source utilities to provide a low-cost computer-controlled product and distributing a program on hardware that blocks execution of modified software have proven contentious issues. Running commercial Web services using open source software without releasing source code has also caused consternation in some quarters.”¹⁸

The new GPL is designed to curtail this practice, and the open source community has now shown its willingness to enforce these license restrictions even against larger companies, such as Verizon. While three of the four lawsuits have been settled, the actions taken by the Software Freedom Law Center have served as a shot across the bow of for-profit companies seeking to leverage inexpensive OSS for commercial gain.¹⁹ As a 2007 Open Source Think Tank Executive Summary Report states: “There are too many open source licenses, but not a single one adequately meets the needs of commercial open source vendors and customers.”²⁰

Convergence and a Hybrid Approach

The OSS debate is most often viewed as being a zero-sum competition between OSS and CS. Recently, however, there has been an increasing recognition that the open source and commercial provider models have begun to converge. For their part, open source providers have been driven by customer demand to adopt several aspects of the commercial model, most significantly “the pursuit of profit”, which includes “generating meaningful revenue and profit from sale of licenses, support, subscriptions and professional services.”²¹ Some open source providers have also built “enterprise-level support organizations.”²²

Similarly, many of the principles upon which OSS is based—collaborative lower-cost development, faster development cycles, and the use of collaborative tools—have been adopted and adapted by traditional providers. Some organizations, such as IBM, have actually begun to fully adopt OSS systems such as Linux into their offerings. Others, such as Microsoft, have engaged the OSS community in a fashion that tries to bridge the gap between OSS and CS. This latter evolving hybrid approach is of the most interest to this study as it provides a telling case study of how even the most well-known commercial providers are attempting to combine ‘open’ principles with the benefits afforded by a central commercial entity. Three central elements of this approach are:

¹⁷ Walsh, Edmund, *Open Source Software Shows Its Muscle*, 03 June, 2008, <http://www.law.com/jsp/legaltechnology/pubArticleLT.jsp?id=1202421869652>

¹⁸ *Ibid*

¹⁹ *Ibid*

²⁰ 2007 Open Source Think Tank, Executive Summary Report

²¹ 2007 Open Source Think Tank

²² *Ibid*

- Data portability
- Support for industry wide standards
- Open connections

Previously, Microsoft provided documentation of its Application Programming Interfaces (APIs) through its Microsoft Developer Network, which is freely accessible online. Microsoft has published Windows and Windows Server protocol APIs and associated documentation free of charge. That information has been offered with the Microsoft promise to encourage, rather than inhibit, third party implementations that make use of the APIs and associated protocols.

Additionally, Microsoft is currently engaged in providing the same level of transparency for all of its high volume products. Accompanying elements of this sort of strategy include an effort to promote and use open data formats. These will either take the form of industry standards, or Microsoft formats that will be transparent and allow royalty-free implementation by third party developers. The claimed advantage of the Microsoft approach lies in its ability to harness some of the benefits of OSS development, which includes modularity and adaptability born of transparency, while also still maintaining and utilizing the support infrastructure and focused resources provided by a large commercial entity. Essential to such an approach is broad engagement across the software development community to ensure compatibility and interoperability with software developed in both OSS and CS environments.

A necessary component of any hybrid approach is a mechanism that provides end users, particularly in defense environments, a level of transparency and feedback to address concerns related to the source code itself. Microsoft's approach was the 2003 establishment of the Government Security Program (GSP) allowing select educational, government, and enterprise customers to audit Microsoft source code. The program includes elements which allow governments unfettered access to source code so Microsoft may then actively address identified concerns.

The last element of a hybrid approach must address the issue of ongoing support and commitment; in the context of defense usage a piece of software may be utilized for years, even decades at a time. That requires a necessary commitment on the part of a software provider to adhere to the same open principals under which software was developed for the entirety of both the software's and derivative software's life spans. In the case of Microsoft, the continued update of file format, patent, and API documentation and architecture represents an institutionalization of hybrid principals.

In addition to these measures, Microsoft has also engaged open source providers through partnerships in order to foster open source development. For example, Microsoft has partnered with JBoss and Novell, as well as MySQL, a direct competitor in the database program market to Microsoft's SQL Server. The partnership has allowed developers to better integrate MySQL into the Windows environment in two important ways – first, by increasing compatibility of MySQL within Windows and second, by allowing MySQL to be integrated into programs developed using Microsoft Visual Studio. Microsoft also launched Port 25, a website devoted to discussing interoperability between Linux, Unix, and Windows operating systems.

The Open Source Debate in the Military Context

The debate outlined above has mostly taken place in the commercial context: corporations or individuals seeking the most effective, secure and cost-efficient software approach for their specific needs. In the last five years, the debate has spilled over into the public sector as governments have increasingly examined the utility of OSS based-systems in meeting a wide range of public sector requirements. Among the more central areas of interest to governments around the world is the appropriateness of OSS in the emerging military context. This debate is in its early stages as governments around the world evaluate the ability of OSS to meet their specific military requirements, particularly as it regards the interlocking issues of security, supportability, risk, resilience in restricted communities, total cost of ownership, and pedigree. While some of the themes and arguments that have emerged in the commercial context regarding the effectiveness of open source versus CS have relevance in this new environment, it is also clear that military and intelligence communities throughout the world have their own sets of requirements and criteria for evaluating all software alternatives.

Trends Driving Software Procurement and Use in the Military

The international security environment has undergone profound and persistent changes since the Fall of 2001. A complex global security environment has emerged that is marked by a range of diffuse, but increasingly coherent, challenges. The proliferation of advanced military and dual-use technologies to more and more actors has allowed for the growth of a wide spectrum of more capable adversaries and actors, compelling militaries to meet a disparate set of military contingencies. In this context, two key trends have emerged amongst many modernizing militaries around the world to meet these evolving operational military requirements. First, the movement toward Network Centric Warfare (NCW) and Network Centric Operations (NCO). A brief discussion of key elements of NCW and NCO is included in Appendix I.

The second key trend in the emerging C4I environment is the increased emphasis on the purchase of “Commercial-Off-The-Shelf” (COTS) technology by many militaries to help enable the development and deployment of NCW systems. The 1990s saw a shift away from proprietary development of hardware requiring specialized software in military technology acquisitions in favor of the concept commonly termed COTS acquisition. Under the previous model, systems and platforms were often locked into cumbersome development/support contracts due the “knowledge-walls” resulting from a dearth of collaboration/established standards across defense contractors. The newer COTS concept sought to address the situation and decrease development/integration time by favoring solutions utilizing established products available in the commercial sector, which could be easily integrated into a single offering.

The rapid prototyping and fast fielding enabled by COTS strategies have been accompanied by related obstacles and concerns. Between 1995 and 2000, Jane’s Information Group noted a 70% increase in the proportion of project budgets defense contractors spent on out-sourcing.²³ That spending increase highlighted the increasingly collaborative nature of defense projects. Along with increased collaboration, issues of product life-cycle became apparent. In 2000, the typical life-cycle of a COTS-based solution was two to three years, which conflicted with the 25 to 40 years of sustainability required of typical government programs.²⁴ Products thereby needed to have clear upgrade paths easily accepting technology insertions at various points

²³ “The Evolution of COTS in the Defence Industry,” *Jane’s Defence Industry*, 1 Oct 2000.

²⁴ *Ibid.*

during their usable life. In terms of software, decreasing product development timelines necessitates modularity to facilitate upgrades driven by the ever-accelerating nature of technological advancement. Additionally, increased collaboration among contractors ('off the shelf usage') has emphasized the need for defense software to be transparent enough to easily facilitate the integration of products (hardware and software) from disparate development paths.

Presence of and Emerging Attitudes Toward OSS in Government and Military Environments

Investigation of OSS penetration within the international C4ISR software market suffers due to definitional inconsistencies across countries, industries, and agencies which all lack a unified conception as to what constitutes 'open source'. There are also no pervasive systems and processes to track OSS use in militaries throughout the world. Jane's, however, identified 47 countries which currently employ C4ISR systems utilizing UNIX, Linux or hardware running on a mixture of both (a breakdown of these countries can be found in Appendix G). Understandably, UNIX use is the most prevalent, which can be attributed to its long history of specialized application. However, UNIX-based systems are increasingly being transitioned to newer Windows or Linux-based operating systems during life-cycle upgrades. A current example would be the U.S. Global Command and Control System (GCCS) utilized by the US and several NATO allies. Currently the system runs on UNIX and is in the process of a transition to a Windows /Java Web environment. Experts interviewed for this paper reported an increasing trend towards developing C4ISR systems independent of software operating systems that leave the decision of running Linux or Windows up to the end user.

One consistent trend in attitudes about OSS procurement is that some national and even local governments seem to have embraced OSS operating systems, particularly in 2003 – 2006, for a variety of public sector purposes. Countries such as Brazil and Germany passed laws mandating that public agencies adopt OSS or Linux specifically, while the French and Italian legislatures are in on-going discussions over bills that forbid anything but OSS.²⁵ Other states—again, Germany—have offered discounts and tax savings for adoption of OSS over commercial alternatives, and still others have made bold declarations about OSS use.

And while interest in OSS in the public sector certainly persists, some of the enthusiasm for OSS has diminished over time. For example, a 2007 Government Open Source Policy survey, the Center for Strategic and International Studies (CSIS) identified 268 open source government policy initiatives worldwide that included "explicit statements" of government policy on the use of open source, ranging from policy mandates to policy preferences and advisory policies.²⁶ The survey revealed that 177 (66%) of these wide-ranging guidance statements were approved, out of which only 6 policies (3.4%) were mandatory policies.²⁷ Thus, increasingly governments are seeking to provide a framework to consider OSS, but there is little appetite for mandating its use on a large scale.

Michael Warrilow, an analyst with Hydrasight, an IT industry analysis firm focused on enterprise software and services in the Asia-Pacific region, noted in April of 2008, "Open source hasn't revolutionized the world

²⁵ Comino and Manenti, p. 6

²⁶ James A. Lewis, Introductory Note, "Government Open Source Policies," *Center for Strategic and International Studies*, August 2007

²⁷ *Ibid*

the way we thought it would five years ago. Three years ago it was much more fashionable.” Mr. Warrilow continued, “Governments were talking about mandating the use of open source . . . but they have walked back from that due to the support bucket.”²⁸ For example, Mr. Warrilow points out that Malaysia was planning on widespread implementation of OSS—it was “going to be everywhere”—but has since come off this stance, due to concerns over the supportability of OSS.²⁹ In addition, Singapore, where the Ministry of Defence ceased the use of Microsoft Office in 2004 and the government has offered tax breaks to companies that use the open source Linux operating system instead of Windows, made a S\$1.3 billion procurement award to a team led by oneMeridian to provide a Standard ICT Operating Environment (SOE) for the public sector in February of 2008. This SOE will be implemented in phases for 74 government agencies, excluding the Ministry of Defense and its agencies, which have developed their own systems, and the Ministry of Education. The oneMeridian consortium includes Microsoft, and the SOE will be based on Microsoft Office.³⁰

Moreover, what interest and enthusiasm that has been developed within governments throughout the world, has not necessarily easily translated to militaries. In fact, Jane’s analysis of OSS in militaries around the world reveals a wide range of attitudes about and approaches to OSS, from notional acceptance and ad hoc implementation; to uncertainty and circumspection; to rejection and resistance; and, as described in the case study of the United States Department of Defense below, elements of all three.

The US DoD is an anomaly in many ways. With the largest global military budget and the broadest and most urgent operational requirements, the United States has been assertive and aggressive in the transformation of its military force to a more network-centric model. OSS is and has been a part of this transformation, although at what level of the technological hierarchy and with how much enthusiasm OSS will be embraced in both the short and long term has yet to be determined. For example, in 2006, the DoD released the Open Technology Development Road Map Plan. This document did not, according to two highly placed individuals involved in charting US DoD technology policy over the last decade, “have much of an impact”³¹ in a tangible sense. However, the document did reinvigorate the debate about OSS in the military environment, and provide top-cover for “on-the-fence Program Managers and Program Executive Officers”³² who were uncertain of what to make of OSS solutions and their relevance and acceptability in the highly classified and high stakes context of emerging C4I systems and platforms. And while no DoD resource is officially tracking use of DoD OSS, Jane’s primary source research with former DoD officials indicates that OSS is broadly used in DoD and will continue to be used at least in an *ad hoc* manner, whether the DoD or US government formally either knows of or acknowledges the degree of use.³³ Furthermore, a former senior government official familiar with US DoD software procurement told Jane’s, in terms of reliability and security, some in DoD believe that

²⁸ Tung, Liam, *Open Source Barred From Australian Government*, ZDNet.com.au, 01 April 2008.

²⁹ *Ibid*

³⁰ Singapore Ministry of Finance Press Release, *SOEASY TO WORK AS “ONE GOVERNMENT”*, 28 February 2008, http://app.mof.gov.sg/news_press/pressdetails.asp?pressID=304

³¹ Discussion with Dr. Philippe Loustenau, May 10, 2008 and verified through discussions with a former Senior US DoD Technologist from May 15-22, 2008.

³² Interviews with a former Senior US DoD Technologist, from May 15-22, 2008.

³³ *Ibid*

purchasing software based on OSS (such as Linux) from a top tier firm with perceived robust support services like IBM or Red Hat represents a similar value proposition to buying software from Microsoft.³⁴

In addition, Jane's primary and secondary source research indicates that OSS has been recently acquired as part of high profile programs and is being considered for key C4I procurements. For example, the US DoD opted to employ IBM BladeCenter Servers that operate a modified Linux distribution for the DDG 1000 destroyer, though this program was canceled in July of 2008 for issues unrelated to software procurement.³⁵ Also, the US Navy CANES program (Consolidated Afloat Networks and Enterprise Services), the next generation IT for afloat systems and by definition integral to Navy C2, is seriously considering applications of what one former DoD official termed the LAMP (Linux, Apache, MySQL, PHP/Perl) stack in order to keep program costs down. The CANES Program is a high profile program; it could be that this program serves as a bell-weather for future attitudes toward OSS in the US DoD environment.³⁶ However, currently, these attitudes are still forming, with pockets of acceptance, resistance, and uncertainty throughout the department. Some of this current uncertainty about OSS is also due to a lack of clarity at even the highest echelons of the DoD regarding key issues toward OSS, such as certification and how prevalent its use is across the organization.³⁷

Assessment of Software Solutions in the Emerging Military Context

Jane's has identified a range of key factors that tend to influence decision-making on software procurement issues in militaries that are transforming to a more network centric force structure and technology base. These factors are not at play equally in every competition or procurement decision in every country throughout the world. In fact, as is discussed in more detail below, the specific requirements of a specific military system or platform (the nature and priority of its mission, its role, the required timing of its deployment); the level of risk aversion or risk acceptance within the culture of given militaries or services; the existing procurement environment and past procurement history both generally and for software; and the clarity of department or ministry-wide guidance on OSS all play important roles in determining whether OSS or CS is preferred in any given software procurement initiative.

Security: The first and most important metric for determining suitability of OSS, CS, or other software alternatives in the military context is the degree of security that various software options can provide. In order to be considered secure in an intense environment in which software is always being probed for vulnerabilities, software must:

- Be able to operate dependably in the most hostile of environments
- Not contain any exploitable vulnerability or weaknesses, and no malicious logic;
- Be able to resist most known attacks and as many unknown attacks as possible;
- Be able to minimize damage and recover quickly when breached.³⁸

³⁴ *Ibid*

³⁵ *IBM and Raytheon Deliver Technology Solution for DDG 1000 Next Generation Navy Destroyers*, IBM, <http://www-03.ibm.com/press/us/en/pressrelease/21033.wss>

³⁶ Email interview with a former Senior US DoD Technologist, May 15, 2008.

³⁷ Interviews with a former Senior US DoD Technologist, from May 15 – 22, 2008.

³⁸ *Ibid*

Proponents of OSS argue that the openness of OSS provides enhanced security to CS providers. This argument rests in large part on two axioms upon which the theory supporting OSS is firmly based:

- Saltzer and Schroeder Open Design Principle: Security must not depend on attacker ignorance.³⁹
- Linus’s Law: Given enough eyeballs, all bugs are shallow.⁴⁰

In an open community, software code is under constant peer review by developers (and, through feedback loops, users) worldwide. Because OSS code is visible to everyone in the community, bugs, be they design flaws or malicious code, can be detected and fixed far more quickly than bugs found in CS. In this context, even some non-catastrophic attacks on software can be seen as part of the security vetting process. As one industry observer noted at an April 2008 conference on Open Source and the Military, some OSS developers “even view the Cyber-Terrorist as part of their QA (Quality Assurance) team.”⁴¹ This is perhaps OSS advocacy taken one step too far and into the realm of near tautology—a system’s invulnerability is enhanced and even proven by its having been breached— but does reflect the strength of belief among OSS advocates that openness creates enhanced security.

End users of software surveyed for this paper, though, have voiced a range of views about the relationship between openness and security. One IT professional working in the US national security environment noted that OSS is “a bit of a headache” for “IT people” in large part because while increasing the number of individuals reviewing the code is accepted as a positive development, allowing *everyone* who wants to view the source code creates challenges and uncertainty in military IT communities. Security of OSS relies, at least in part, on the users and developers remaining ascendant in the dynamic and on-going competition between those seeking to enhance security and those seeking to exploit vulnerability. The fact that open source code can be easily read makes it possible for attackers to become intimately familiar with an OSS component’s implementation.⁴² Such familiarity can lead to sophisticated and irregular attacks by adversaries who are increasingly capable, clever, motivated, and technologically savvy, though OSS advocates do point out that some widely-used OSS does incorporate code reviews, which do have varying degrees of security focus (Mozilla offers the Mozilla Security Bug Bounty Program, for example).⁴³ Still, concern exists about these reviews. As discussed earlier in this paper, some observers believe that security audits of open source code is undermined by a lack of full understanding of how much code is reviewed and by whom.

Moreover, some end-users interviewed for this paper believed that the general ubiquity of Microsoft’s software has made it more “open.” As one senior executive with a firm that provides secure and formal messaging to defense, aviation, intelligence and other government agencies reported, “Microsoft is just deployed everywhere.” Because of this widespread use and deployment, “everyone has had a go at it and more people are finding bugs in it.”⁴⁴

³⁹ Charles B. Weinstock, *Thoughts on Trust and Vulnerability in Open Source Software*, Presentation to the Military and OSS Conference, April 21, 2008.

⁴⁰ Jarzombek and Goertzel

⁴¹ Weinstock, slide 12.

⁴² Weinstock, slide 11.

⁴³ Jarzombek and Goertzel, slide 13.

⁴⁴ Interview with industry executive in Alexandria, VA on June 17, 2008.

In the more security-conscious military environment and given the specific requirements of the emerging operational environment, it is not clear whether OSS or traditional CS is the most secure option. In fact, Jane's research indicates that OSS is neither *any more nor any less* secure than proprietary software. One former senior technology officer in the US Department of Defense told Jane's that a study commissioned by his / her office refuted the notion that open source software was more "secure" and determined that OSS is not more nor less secure based on analysis of associated vulnerability data. Similarly, even some strong advocates of OSS assert that while OSS has an underlying principle of openness that should create conditions for enhanced security, neither "OSS nor proprietary software is always more secure."⁴⁵ In addition, while OSS and CS are often seen as two ends of a broad spectrum for approaches to software development and maintenance, other alternatives combine the advantages of the scrutiny and openness of open source with the perceived reliability and accountability of CS. For example, IBM or Red Hat's use of OSS is seen by some in the US DoD as an effective compromise. Similarly, Microsoft's hybrid approach in which code is made available to users for review, but not for modification, also has the potential to create security advantages that may not be realized by OSS or CS alone.

Supportability: As referenced earlier in this section on OSS and CS in the military environment, some of the enthusiasm for OSS in public sector and military use has waned since 2006, in large part due to the "support bucket"; that is, the ability of the provider to deliver timely and effective support and service to the software over the course of its life-cycle, which in a military context can be two or three decades rather than two to three years. CS providers frequently point to their ability to provide enhanced and tailored support to their military clients as one of the most important advantages of CS. In addition, some militaries that are being particularly aggressive and thoughtful in their approach to developing and implementing NCW-related systems and platforms have cited supportability as a key factor influencing their preference for CS.

For example, in January 2008, the Australian Government Information Management Office (AGIMO) released summary findings of a survey on the use of OSS in government agencies that was conducted in February and March 2007. Most Australian government agencies expected that OSS usage would increase over the next five years, citing the freedom to choose vendor support providers and the greater degree of perceived flexibility associated with OSS. However, the survey also revealed many challenges faced by Australian government agencies in the adoption of OSS, most notably vendor support. Another challenge is the difficulty in testing OSS solutions, which is important in the military context.⁴⁶ "According to the CIO's of Australia's three largest government departments -- Defence, the Australian Tax Office, and Centrelink -- support is a very real concern -- the central reason why more open source is not widely used in government."⁴⁷

Resilience and Restricted Communities: A fundamental element of the purported value of OSS is that its openness allows for the speedy detection of design flaws or malicious code as well as the speedy development

⁴⁵ Wheeler, page 8

⁴⁶ Rodney Gedda, "Government agencies view open source as positive: survey," *Computerworld Australia*, 21 January 2008

⁴⁷ Liam Tung, "6 months vulnerability report for operating systems: Open source barred from Australian government," *ZDNet.com.au*, 1 April 2008; Jones, Jeff, Windows Vista – 6 Month Vulnerability Report, http://blogs.csoonline.com/windows_vista_6_month_vulnerability_report, 21 June 2007.

of patches designed to defeat these specific bugs. As a result, OSS operating systems are known for releasing patches “early and often.” However, Microsoft, in a Windows Vista Six Month Vulnerability Report posted online in June of 2007 by a Microsoft employee, suggests that Windows Vista shows a trend of fewer total and fewer High severity vulnerabilities at the 6 month mark compared to Windows XP and to other modern competitive workstation operating systems, including open source operating systems.⁴⁸

However, it is not fully clear that the strategy of releasing early and often is particularly desirable or useful in a military and security environment. Military networks, including classified military networks, are generally closed with no outside access, particularly to the internet. The closed nature of these networks makes it more difficult, but certainly not impossible, to apply software updates as they become available. Given the increased frequency of OSS updates, this can, as one US government IT professional noted, be problematic⁴⁹ for IT providers if there is a compelling and persistent need to have the most recent update. Jane’s primary and secondary source research reveals that “Waiting for major releases (1.4 to 1.5 rather than 1.4 to 1.4.1 to 1.4.2, etc) is generally acceptable”⁵⁰ in a military environment where regular interruptions of service to apply fixes that may not be required is simply not necessary.

In addition, for OSS to be most effective it requires reasonably large communities with sufficient volunteers to scrutinize the code and develop modifications, which are reported back to the user community. Observers have repeatedly noted that a role for OSS in “high side” operations exists and will continue to. However, considerable challenges also exist. Most notably, attracting a sufficient pool of IT professionals to the military environment in a highly competitive commercial market is far from a foregone conclusion. Even if a sufficient pool of individuals can be brought together, maintaining that pool to work on the code in the traditional OSS manner creates another key challenge.⁵¹

Familiarity, Ease of Use and Deployment: Closely related to supportability of software in the military environment is the ability of military personnel to actually use and implement the systems with which they have been supplied. Military technology and operating environments both evolve rapidly, and militaries rarely have the luxury of an abundance of time to deploy emerging technologies to the field or to train soldiers, sailors, airmen, Marines or intelligence community officers and operatives how best to interface with these new technologies. Thus, developing operating systems that are relatively easy to deploy and that service-men and women are likely to have some familiarity with is an increasingly important driver in the complicated defense software procurement process.

For example, in 2003, the Royal Netherlands Army (RNLA) created a range of “state-of-the-art” situational awareness and Command-and-Control (C2) support tools based on a Microsoft Windows Server System. Among the many factors considered by the RNLA was the easy to use and manage infrastructure, the

⁴⁸ Jones, Jeff, Windows Vista – 6 Month Vulnerability Report, http://blogs.csoonline.com/windows_vista_6_month_vulnerability_report, 21 June 2007.

⁴⁹ Interview with a USG IT professional in the national security community on OSS, 19 May 2008.

⁵⁰ *Ibid*

⁵¹ These insights are taken from a number of open secondary sources (Weinstock, Wheeler) as well as interviews with members of Jane’s expert network, including former senior US DoD defense technologist, current government IT professionals and European defense industry observers.

compatibility and flexibility associated with commercial-off-the-shelf products, and the lack of training required to become familiar with the tactical messaging systems (TMS). As Colonel Geerlof Kanis, Commander, Command and Control Support Center (RNLA), stated: “To make use of state-of-the-art technology, all the products and the infrastructure itself are based as much as possible on commercial off-the-shelf software and hardware using open industry standards. “We don’t invent technologies on our own; we buy products on the market and use them to assemble military systems,” says Kanis. “And using industry standards opens up the full range of commercial off-the-shelf products that become available...,” Kanis continued, “The thing we really wanted from Microsoft technology was an easy to manage infrastructure.”⁵² Microsoft did not, at least at the time, offer a ‘perfect’ solution for the RNLA, but the solution they did offer was both highly effective and preferred to other software options, specifically in terms of the ease of use of the software and the high degree of familiarity the RNLA software engineers and operators had with, and thus the lack of training required to operate, Microsoft systems. As Colonel Kanis again noted, “the alternative would have been something far too difficult to use in the field.”⁵³

Similarly in 2003, Israeli defense company Elbit began development of the next generation of C2 technology for their UAV products. After an initial evaluation period involving the Israeli Defense Force (IDF), Elbit decided to utilize Microsoft products for their entire C2 hierarchy, from the UAV itself to the individual control stations. Elbit entered a strategic partnership with Microsoft in which Microsoft would develop a platform consisting of Windows, the .NET Framework, and SQL Server, on top of which Elbit would install proprietary C2 software. In evaluating whether to use OSS or CSS for their C2 architecture, Elbit weighed a number of key factors, many of which were related to ease of use and integration as well as supportability. First, as an integrator of systems, Elbit was primarily concerned with ease of integration among different layers of the C2 hierarchy. Second, Elbit was concerned with the costs associated with having to hire and / or train in-house engineers familiar with OSS software. Third, Elbit wanted assurances that their product could be supported throughout a long life-cycle and, ultimately, the fact that Microsoft could provide multiple products with single-vendor convenience and an integrated development environment was extremely attractive to Elbit, in terms of costs, supportability and ease. Elbit also felt that Microsoft could provide the best life-cycle support for their product. Operationally, Elbit wanted to avoid using multiple architectures throughout its C2 network. Because end-user workstations would be running Windows, Elbit felt it was an additional advantage to run Windows on both servers and the UAV. As of 2008, the project is nearing completion, with the Israeli Defense Force remaining actively involved in development and testing.⁵⁴

Politics and Risk Aversion: Context and culture are also key drivers of military procurements of all kinds, including software procurements: What is the history of software procurement programs? Of defense programs more generally? Is there a preferred provider or incumbent? What are the budgetary constraints? What is a given military’s attitude toward risk? All of these are key questions that are likely to be as important, or at least of exceptional importance, as the issues identified above in any major defense software procurement.

⁵² Microsoft Windows Server System Customer Solution Case Study, “Living Architecture Brings Command and Control Into the Information Age,” Microsoft Corporation, 2005.

⁵³ *Ibid*

⁵⁴ All information in this paragraph taken from an interview with Elbit representative involved in the program, June 4, 2008.

For example, the 2002 decision to opt for ‘Windows for Warships’ in preference to an Open Source solution on the Type 45 destroyers, and eventually the whole of the UK submarine fleet, was heavily influenced by existing practice within the British MoD and its designated contractors—primarily BAE Systems Insyte which had standardized on Windows 2000 and did not consider other alternatives—as well as political environment that reinforced a strong tendency toward risk aversion.

The Type 45 CMS decision was taken at about the same time (2002-3) as the UK National Audit Office (NAO) was conducting a review of licence fees. The NAO reported that the MoD was at that stage spending around £60 million each year on software licences but hoped to achieve savings on the order of £17 million over a four year period through a recently negotiated agreement with Microsoft. Thus, this decision was also a function of evolving MoD policy and a preference for Microsoft in order to bring order and reduced costs to over 1,100 licenses and associated fees. In addition, the UK MoD software procurement policy was then and continues to be heavily influenced by a series of programs that were delayed and over run on cost because major contractors (GEC-Marconi, for example), failed to make appropriate ‘up-front’ investments in the development phase. This had been identified (some years before the 2002 report) as a major failing of software-intensive programs and had drawn unfavourable comment from at least one parliamentary committee. Purchase of proven ‘Off the Shelf’ technology was seen as a low risk option – entirely consistent with a procurement policy that sought to transfer a higher proportion of technical and commercial risk to industry.

In the case of CMS options for the Royal Navy, Microsoft represented a safer option given their size and the commercial availability/maturity of their solution. Interestingly, the persistent preference for Microsoft products in elements of the MoD is taking place within the context of a general receptiveness to OSS in other parts of the UK government. In 2004, a document published by the UK Cabinet Office elucidated a government policy mandating enhanced consideration of OSS alongside CS and diversification of software suppliers in order to achieve overall best “value for money.”⁵⁵ The Windows for Warships decision reveals that defense software procurement decisions will be influenced, sometimes heavily, by history, context and the culture of the military that is procuring the software, specifically their propensity for risk aversion. In many cases in the military context, which is often marked by a strong aversion to risk due to the high costs (lives) of failure, a risk averse procurement officer or decision-maker will be more likely to seek out providers with long histories of providing software and operating systems to both the commercial and public sector—such as Microsoft, but also others including Red Hat or IBM leveraging OSS—and thus are “known quantities.”

Pedigree and Trust: The perception that traditional CS providers are more reliable and more “trustworthy” stems from a reasonably pervasive uncertainty and lack of guidance about OSS—strengths, weaknesses, degree of technological maturity, etc—within a number of militaries surveyed throughout the world, even in militaries that are currently using OSS-based operating systems. Certainly OSS is being used as part of C4I platforms in militaries throughout the world, including the United States, the United Kingdom, Australia, and

⁵⁵ *UK OSS Policy*, Office of Government Commerce, http://www.govtalk.gov.uk/documents/oss_policy_version2.pdf

many others. However, there has been little in the way of a concerted, coordinated movement or vision in these countries that elucidates a department / ministry wide (or even service wide) approach to OSS. One former senior technology officer in the US DoD suggested to Jane's that while OSS is being used in a more or less ad hoc way across the US DoD, there are still some fundamental questions about certification, and how OSS is best used in restricted use and highly classified environments. This is not to suggest that there is a universal bias against OSS, but rather that some government officials in general and the military more specifically are simply unsure how best to view OSS in this context, or have yet to have their concerns and questions regarding OSS appropriately answered.

This uncertainty has reinforced many of the pre-existing perceptions that CS and CS providers are considerably *less risky* than OSS, particularly as it regards the development process.⁵⁶ Did the OSS development team monitor each developer's initial contribution or only his / her later modifications and updates? What were the requirements, including security requirements that the original OSS was built to satisfy?⁵⁷ In a military environment such questions do not linger in the back of procurement officials or Program Managers or decision-makers' minds. They tend to loudly bang against the most sensitive and most exposed areas, and drive procurement officials to seek the "safe refuge" and *known pedigree* of trusted providers. This was, at least in part, the reasoning behind Elbit's decision to engage Microsoft from the outset of the development of its unmanned aerial system as well as the announcement of the United Kingdom's 'Windows for Warships' procurement discussed above. Clearly, this could change with the procurement of OSS to support particularly visible programs (CANES in the US), but even if baseline concerns are allayed—i.e. there is a systematic and acceptable approach to OSS and coherence of action to implement this approach—the issues of supportability, development, risk, security, and resilience will have to be weighed for each individual procurement.

Ubiquity: Much of the support for CS solutions uncovered by Jane's, particularly Microsoft solutions, revolves around the "ubiquity" of the software. Microsoft software is "everywhere", which means that: "nearly all military personnel entering service are already knowledgeable on how to use Microsoft;"⁵⁸ more people are using the software and thus are uncovering bugs that are reported back to Microsoft; there is a mature dedicated support system, as well as established partners and system integrators and a brand name that can be called upon to support the general risk averse military users. In addition, some users report that the general wide-spread use of Microsoft has allowed Microsoft to improve its offerings over time. As one defense industry user stated, "Fifteen years ago we might not have chosen Microsoft, but now the software is ubiquitous and efficient."⁵⁹

Total Cost of Ownership: The calculation of total cost of ownership of software, like many procurement costs, cannot be understood in terms solely, or even in the military context, primarily, of the initial cost of acquisition.⁶⁰ As discussed at the outset of the paper, CS systems typically have considerably higher upfront costs and periodically impose upgrade costs. However, OSS—despite once being known as Free (Libre)

⁵⁶ Jarzombek and Goertzel

⁵⁷ *Ibid*, slide 19

⁵⁸ Interview with Sr. Executives of Boldon James, a QinetQ company

⁵⁹ *Ibid*

⁶⁰ Wheeler

OSS—is not free either. In fact, Jane’s research indicates that some, though not all, OSS providers are increasingly attempting to monetize OSS offerings, which requires the implementation of different business models that may mitigate or eliminate the supposed lower up-front cost of OSS. Red Hat, for example, charges up-front fees per CPU model that are similar to those charged by CS providers: Basic Linux Enterprise subscriptions start at roughly \$350, including web support and 2 business-day responses, up to premium subscriptions for \$1,300 per year.⁶¹

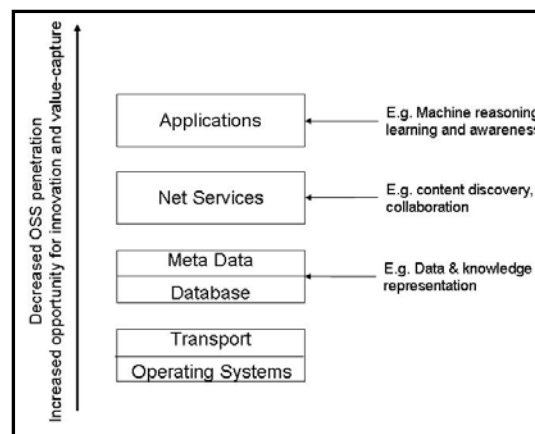
Of course, not all OSS providers follow this up-front per-CPU costing model. Even those that do have lower up-front costs are not free either, though costs for OSS are most often paid in “time rather than money.”⁶² Among the more significant of these costs are transition, downtime, usability, familiarity and training. Transition costs refer to the costs of moving from a traditionally CS offering to an OSS offering. Because many proprietary systems are not compatible with comparable OSS systems, the cost for transitioning large numbers of users from CS to OSS when CS licenses expire can be considerable in terms of the time it takes to train a large number of users on a new system. This cost is increased by the widely-held perception that most OSS solutions have weak graphical interfaces—relative to popular, commercially available interfaces,—that require more time to learn to effectively navigate. This ease of use and deployability was cited by the RNLA as the primary reason for choosing Microsoft solutions for their C2 architecture. In that case, cost was understood not in terms of time or money, but rather in terms of operational preparedness and effectiveness. If another solution was acquired, the RNLA believes it would have not been as prepared to carry out its missions within the necessary time frame. Thus, in a military context the concept of total cost of ownership (TCO) includes not only the cost to procure OSS-based operating systems, but also the time to train personnel to use this system, and the loss of preparedness and operational capability that *may* occur if unfamiliarity with OSS delays the deployment of systems and platforms.

Emerging Requirements and Levels on the Technological Hierarchy

The section above detailed a general assessment of OSS versus CS and other software alternatives and their relative appropriateness in the emerging military environment. Much of this assessment is focused on the utility of each software approach as the foundation for operating systems. In this section, Jane’s will evaluate the utility of CS and OSS at various stages of technological hierarchy associated with emerging C4I requirements. To do this, Jane’s will further examine the US DoD and its emerging C4I requirements.

The Technology Hierarchy and Emerging C4I Requirements

Analysts and observers of the US DoD IT community, and developments that Jane’s accessed for this project provided an overarching framework for understanding the hierarchy of C4I-related technology and requirements. This hierarchy of technology has four discrete layers listed in the chart to the right. As the DoD, and, frankly, other militaries, continues its



⁶¹ Red Hat Costing Sheet

<https://www.redhat.com/wapps/store/catalog.html;jsessionid=lvhgbGcqyhQZcfCQPWLn6eFqd75vnQoAyC.www0>

⁶² *Ibid*, page 6

transformation and implements and grows more net centric technologies and concepts, the DoD will continue to invest in increasing: communications bandwidth, the number of people connected, the number and kind of devices on the network, the number of systems and applications, quantities of information, types of media.

The DoD strategy for navigating the increasing complexity borne of greater access to better information and the increased connectivity of nodes on the battlefield involves four discreet stages:

- *Creation of Meta-Data* in order to tag and represent all the data and knowledge on the network
- *Automation of this tagging* will be essential in achieving the massive scale required by the US military (and, conceivably, for other militaries) in both volume and speed. Operators will require tools for semi-automated, rapid assessment of the quality and relevance of the massive amounts of information available across the network. These requirements call for mega-scale data management to provide robust and scalable knowledge management to support very short decision timelines.
- *Inter-related data from many different media*, for example text, voice, video, model outputs, will also be key. Combining data from one or more sources, one or more once separate formats, or from one time and place with other contexts will be critical. Technologies will need to tag and connect the data from all of these different forms, while maintaining provenance and the original data
- *Finally, understanding and using the data within the context of specific missions will be an important requirement.* Operators will need context-aware presentations that are sensitive to a variety of environmental factors (location, time, and device) as well as to the psychological, perceptual, cognitive, and social characteristics of the user and groups. This calls for human/system collaboration, which results in dramatic reduction in workload for operators under stress and in time to conduct complex analyses characterized by uncertainty and ambiguity.

Tools to meet these emerging and existing requirements have been developed, but not generally, to the robust and scalable levels necessary, creating an opportunity for further competition between OSS and CS solutions to drive innovation at the higher levels of the technology hierarchy. However, experts familiar with DoD efforts to create these tools believe that the maturity of technologies which exist (or will exist) *at the bottom of the hierarchy*, make OSS penetration feasible at that level while software slotted into higher positions of the hierarchy is of narrower applicability and has increasingly specialized development requirements. As one traverses upward, OSS penetration is limited due to the decreasing size of available development communities. There exists a need for greater innovation at the top of the hierarchy (machine reasoning/learning, etc.). The ability (or lack thereof) of developers to capture value from the creation of innovative solutions also contributes to decreased suitability of OSS at the higher levels of the hierarchy.

In short, the DoD is seeking open interfaces and open data formats, and open standards to a certain extent, rather than open technologies. The insistence on open interfaces and data formats will be increasingly strong as the complexity of the enterprise being built will require it. As technologies evolve and their numbers and type increases, open interfaces minimize complexities, reduce cascading software dependencies, and eliminate the need to re-engineer or re-integrate a system when new technology is introduced. The DoD will likely, over time, enforce a ban on proprietary data formats and API's.⁶³ Thus, while Jane's analysis suggests

⁶³ Dr. Philippe Loustaunau

that OSS *can* (and, in fact in some cases, does) play an important role in the lower part of the hierarchy—i.e. operating systems—it is less likely to effectively meet the emerging requirements at the higher end.

Conclusion

The demand for and utility of OSS in the military context, particularly as it applies to C4I systems, is a complicated issue that involves its own decision-making calculus. The on-going debate regarding OSS and CS strengths and weaknesses is not entirely irrelevant, but the high-stakes of the military context—tactical and strategic advantage and ultimately, lives—the long lives of C4I platforms; operational exigencies; and the generally risk averse nature of most militaries have produced priorities for software acquisition that are different than many of those commonly associated with commercial procurements. Clearly, OSS has played and will continue to play some role in the military context, but it is just as evident that trusted commercial providers are sought after in an environment that stresses supportability, ease of use, deployability and restricted communities. Several militaries that are pursuing transition to a more network-centric model of warfare (Australia, the UK, Israel, the Netherlands) have aggressively sought to solidify their relationships with CS and trusted providers even as other elements of their governments have sought to test OSS solutions.

Additionally, some providers are adapting their traditional approaches to supporting the public sector and military C4I communities more specifically. IBM, for example, has embraced OSS solutions at the operating system level, and many US DoD sources interviewed for this paper suggested comfort with this approach. Microsoft has also adapted elements of their approach in the past several years and has increasingly made source code available for review by military clients while also increasing interaction with military clients, particularly within the development stage, as with Elbit, RNLA, and others referenced in this paper. These adjusted and hybrid approaches from both CS and OSS providers—combining increased levels of code and development transparency with dedicated support—have found resonance in the military C4I environment and have demonstrated an ability to address the specific concerns and priorities (trust / pedigree, supportability, ease of use, deferred risk) that are of utmost concern to many military decision-makers.

Appendix A: Trends Driving Software Procurement & Use in Military

<i>Intractable Territorial Conflicts:</i>	The persistence of several seemingly intractable territorial conflicts that have the potential to quickly escalate and spin out of control, as well as the emergence and retrenching of a range of relatively new insurgencies
<i>Coherence and Efficacy of the State</i>	Many states are not as firmly in control over the whole of the territory within their borders, and individuals often have a more intense and relevant identification with transnational ethnic, religious or tribal groups
<i>Proliferation</i>	The attempted proliferation of weapons of mass destruction, particularly nuclear weapons, is an important driver that will no doubt help shape future security environments. So, too, will be the proliferation of advanced conventional military and Commercial Off-The-Shelf (COTS) technologies that will signal an environment in which more actors have more capability. Consequently, it is becoming increasingly difficult to determine the nature of an adversary's capability
<i>Competition for Resources</i>	Demand for secure access to energy is and will continue to be a driver of potential conflict and of future security environments, but so too will be competitions for other key resources, particularly water, but also food. Ensuring the security of access to these resources will increasingly come to influence and shape the national security policies of a range of states
<i>Demographic Shifts</i>	Fundamental shifts in the demographic make-up of societies, due to urbanization, disease, environmental disaster, population movement, and low or high birth rates, will place significant pressure on societies and polities and affect states' ability to devise and carry out security strategies as well as impact the general stability and constitution of particular regions
<i>Globalization</i>	Globalization drives the future security environment in three discrete ways. First, it facilitates the proliferation of technologies to a wider array of actors. Second, the economics of globalization (winners and losers) drive disparities between the economic and political centers of power and peripheries. Third, the increased access to information, usually highly biased or designed to reinforce a particular world view, allows for the globalization of causes, conflicts, ideologies and personalities

Appendix B: Network-Centric Characteristics and Definitions

<i>Agile and Flexible</i>	Networks must be able to respond with pace to the range of contingencies and conflicts that forces may face
<i>Robust and Secure</i>	Although it is impossible to make networks truly impenetrable, minimizing the vulnerability of networks and the systems and software that support these networks is essential. So, too, is making these systems resilient enough to withstand attacks and responsive enough to mitigate attacks
<i>Interoperable</i>	As militaries are increasingly required to serve as part of joint, allied and coalition operations—peacekeeping, reconstruction and stability operations, counter-crime / terrorism / drug network operations, humanitarian relief—the need for networks to be able to enable communication and command across services, coalition partners and non-military organizations becomes essential.
<i>Information to Knowledge</i>	The existing operational environment is exceeding the human ability to adapt and fully exploit technology. Soldiers, sailors, airmen and Marines are often overwhelmed with information and intelligence and simply cannot find, much less use, the information that is most relevant to their existing mission or role. The compressed time frames of the current operating environments magnify this dynamic and create pressing requirements for C4I resources that allow for the easy dissemination <i>and pull</i> of tailored information and knowledge that will meet the specific demands of those that are performing the query

Appendix C: Origins of Open Source Software

The concept of freely distributed computer programs along with available source code became popular as result of two major operating system projects – the Berkely Software Distribution (BSD, also called Berkely Unix) project and the GNU/Linux project. During the late 1990s, the Open Source Initiative (OSI) officially coined the term “open source.”

Open source first appeared in the 1970s when Richard Stallman, an American software developer, resigned from the MIT Artificial Intelligence (AI) Lab and launched his GNU project with the ultimate objective of building a free operating system. Stallman, who believed that sharing source-code and ideas was fundamental to freedom of speech, developed a ‘free’ version of the widely used “Unix” operating system and founded the Free Software Foundation. Stallman wrote a complete operating system and developed an entire set of development tools and applications that he provided to the public, free of charge.⁶⁴

The Unix operating system, developed over the course of the 1970s, was the first operating system that remained independent of specific hardware, which freed users and programmers from the dictates of hardware designers. UNIX could be “ported” to different machines, which allowed the same program to run on completely different hardware.⁶⁵ Open source software continued its development during the 1980s and early 1990s, but between 1991 and 1992 the entire landscape of open source software changed dramatically. The GNU project had either written or located most parts of a complete Unix system by 1991, but it encountered significant problems with its ‘kernel’.⁶⁶ At the same time, an unknown young programmer named Linus Torvalds wrote a revolutionary operating system kernel and distributed it through the internet for free – Torvalds’ kernel (“Linux”) contributed to hundreds of separate projects, many of which were under the umbrella of the GNU project that led to the creation of the Linux operating system.

Both Stallman and Torvalds faced legal challenges to their general public licenses (GPLs). Stallman wrote the original Emacs text editor as part of his GNU project during the 1970s. Stallman’s Emacs was modified in 1982 by James Goslig. Stallman subsequently used Goslig’s source code to develop GNU Emacs at the same time that a company purchased Goslig’s Emacs. The company asserted that Stallman was not authorized to distribute GNU Emacs from the new copyright owner. This led Stallman to create Emacs General Public License (GPL) in 1988 – later renamed GNU General Public License in 1989. Stallman challenged the idea of a ‘copyright’ by establishing the concept of *copyleft*, described by the GNU project below:

“*Copyleft* says that anyone who redistributes the software, with or without changes, must pass along the freedom to further copy and change it. Copyleft guarantees that every user has freedom. *Copyleft* also provides an incentive for other programmers to add to free software. Important free programs such as the GNU C++ compiler exist only because of this. Copyleft also helps programmers who want to contribute improvements to free software get permission to do that. These programmers often work for companies or universities that would do almost anything to get more money. A programmer may want to contribute her changes to the community, but her employer may want to turn the changes into a proprietary software product. When we explain to the employer that it is illegal

⁶⁴ Eric Kidd, *A History of Open Source*, <http://static.userland.com/userLandDiscussArchive/msg019844.html>

⁶⁵ Newman

⁶⁶ The kernel is the central component of most computer operating systems (OS) with responsibilities that include managing the system's resources (the communication between hardware and software components).

to distribute the improved version except as free software, the employer usually decides to release it as free software rather than throw it away. To copyleft a program, we first state that it is copyrighted; then we add distribution terms, which are a legal instrument that gives everyone the rights to use, modify, and redistribute the program's code *or any program derived from it* but only if the distribution terms are unchanged. Thus, the code and the freedoms become legally inseparable.”⁶⁷

Following similar legal challenges faced by Torvalds and the increasing popularity of Linux, open source finally established its credibility within the software industry. Although the free software philosophy was still unclear to many companies and governments, commercial industry interest in open source began to grow.

⁶⁷ *GNU Operating System*, GNU Project, <http://www.gnu.org/copyleft/>

Appendix D: LP Committee License Category Definitions

License Category	Definition
Popular and Widely Used/Strong Community	This category includes open source licenses that are jointly-developed and widely used by the open source community. These licenses originate from the open source community. The LP Committee used statistics from public sources to determine which licenses are widely used. This category also includes some licenses that are widely used within their communities even though they may not be considered the most popular licenses. These licenses, which include the General Public License (GPL) and the Lesser General Public License (LGPL), carry several requirements. First, a copy of the license must be provided to each end user. Second, a copy of the open source code must either be provided to each end user <i>or</i> be made publicly available (either upon request or through an online link where it can be accessed). Third, these licenses do not permit compensation for the open source software, although they do allow companies to earn revenue from fees charged for support, warranties and/or indemnifications.
Special Purpose	Licenses in this category were identified as those that meet the special needs of certain licensors, such as schools and government agencies. These licensors maintain that unique rules for government copyrights associated with the licenses. Since the requirement that source code be provided for modifications only applies if the user intends to redistribute the code outside of its organization, this requirement is not significant for these end user.
Redundant Licenses	Licenses the LP Committee considered to be wholly redundant or partially redundant with existing licenses were grouped in this category. Several licenses in this category, however, have considerable followings but are categorized as 'redundant' in an effort to deal with the license proliferation problem.
Non-Reusable Licenses	Licenses in this group are specific to their authors and cannot be reused by others. Many, but not all, of these licenses fall into the category of vanity licenses.
Superseded Licenses	Licenses in this category have either been superseded by newer editions or have been voluntarily retired. Once retired, this category suggests that these licenses should no longer be used although the LP Committee assumes that some licensors may continue to use them regardless.
Other/Miscellaneous Licenses	Any licenses that cannot be grouped into the other five categories are listed in this category by the LP Committee.

Source: License Proliferation Categories, License Proliferation Report, License Proliferation Committee, Open Source Initiative (OSI), <http://opensource.org/osi3.0/proliferation-report>

Appendix E: Individual LP Committee Licenses by Category

License Category	Definition
Popular and Widely Used/Strong Community	<p>Apache License, 2.0 New and Simplified BSD licenses GNU General Public License (GPL) GNU Library or "Lesser" General Public License (LGPL) MIT license Mozilla Public License 1.1 (MPL) Common Development and Distribution License Common Public License 1.0 Eclipse Public License</p>
Special Purpose	<p>Educational Community License NASA Open Source Agreement 1.3 Open Group Test Suite License</p>
Redundant Licenses	<p>Adaptive Public License Artistic license 2.0 Open Software License Qt Public License (QPL) zlib/libpng license</p>
Non-Reusable Licenses	<p>Apple Public Source License Computer Associates Trusted Open Source License 1.1 CUA Office Public License Version 1.0 EU DataGrid Software License Entessa Public License Frameworkx License TIBM Public License TMotosoto License TMultics License Naumen Public License Nethack General Public License Nokia Open Source License OCLC Research Public License 2.0 PHP License Python license (CNRI Python License) Python Software Foundation License RealNetworks Public Source License V1.0 Reciprocal Public License Ricoh Source Code Public License Sleepycat License Sun Public License Sybase Open Watcom Public License 1.0 Vovida Software License v. 1.0 W3C License wxWindows Library License Zope Public License</p>
Superseded Licenses	<p>Apache Software License Artistic license Eiffel Forum License Lucent Public License (Plan9) Mozilla Public License 1.0 (MPL)</p>
Other/Miscellaneous Licenses	<p>Intel Open Source License Jabber Open Source License MITRE Collaborative Virtual Workspace License (CVW)</p>

	<p>License) Sun Industry Standards Source License (SISSL) Affero GNU Public License Boost Software License (BSL1.0) Common Public Attribution License 1.0 (CPAL) GNU General Public License version 3.0 (GPLv3) GNU Library or "Lesser" General Public License version 3.0 (LGPLv3) ISC License Microsoft Public License (Ms-PL) Microsoft Reciprocal License (Ms-RL) NTP License Reciprocal Public License 1.5 (RPL1.5) Simple Public License 2.0</p>
--	--

Appendix F: Number of UNIX/Linux C4ISR Platforms by Country

Country	UNIX (Dual Compatible C4ISR Included)	LINUX (Dual Compatible C4ISR Included)	UNIX/LINUX (DUAL) Compatible
Australia	3	1	1
Botswana	1	0	0
Brazil	1	1	1
Bulgaria	1	0	0
Canada	5	2	2
Chile	1	0	0
Colombia	1	0	0
Denmark	2	1	1
Ecuador	1	0	0
France	10	7	3
Georgia	1	0	0
Germany	7	1	1
Greece	1	0	0
India	2	0	0
Indonesia	1	0	0
Ireland	1	1	1
Israel	1	0	0
Italy	3	0	0
Japan	2	0	0
Jordan	1	1	1
Kuwait	2	0	0
Malaysia	3	1	0
Mexico	1	2	3
Netherlands	4	1	1
New Zealand	2	1	1
Norway	2	1	1
Pakistan	2	0	0
Peru	1	0	0
Poland	1	2	0
Portugal	1	1	1
Romania	1	1	1
Saudi Arabia	3	0	0
Singapore	2	0	0
Slovakia	1	0	0
South Africa	1	0	0
South Korea	1	0	0
Spain	4	1	1
Sweden	3	0	0
Switzerland	1	0	0
Taiwan	3	1	1
Tunisia	1	0	0
Turkey	1	0	0
UAE	1	0	0
UK	7	2	0
USA	12	7	4
Venezuela	1	0	0
TOTALS	UNIX: 108	Linux: 36	Dual: 25

Appendix G: Approved National Government OSS Policies

GOVERNMENT	BRANCH OR AGENCY	ACTION	DATE	DETAILS & SOURCES
Argentina	National Information Technology Office & National Information Office	Advisory	March 2004	The two institutions, which coordinate IT policy and implementation, announced that they promote Linux in all applications in public administration. The rationale for this decision is lower costs, creating local employment, and security.
Australia	Tax Office	Advisory	Feb.2004	The Australian Tax Office will consider OSS alongside proprietary solutions.
Australia	Information	Management	Office	A document outlines OSS options for government agencies but does not promote OSS; procurement decisions should be made on the standard criteria of fitness for purpose and value for money.
Belgium	Council of Ministers	Mandatory	June 2004	New directives and recommendations approved for use of open standards and OSS by the Federal Ministries. New ICT systems must be based on open standards; new software will have to be delivered with source code and without licensing restrictions, etc.
Belgium	Council of Ministers	Preference	June 2004	Federally commissioned software must be delivered with the source code; federal authorities should try to avoid proprietary software, but should make final decisions based on total cost of ownership.
Belgium	Parliamentary Committee	R&D	Mar. 2003	A Parliamentary committee on the use of ICT in Federal Parliament released a report stressing importance of using open standards.
Brazil	Executive / National Institute of IT	Advisory	Nov. 2003	Govt. initiative urges ministries and other agencies to use OSS, as well as evaluate how IT could benefit from open-software.
Brazil	Executive	Preference	May 2005	Brazil launched PC Conectado, in attempt to sell 1 million low-cost computers, & excluding proprietary software from the project based on belief that free software would spur national industry.
Brazil	Ministries / Executive	Preference	Aug. 2004	Through its Digital Inclusion Program, Brazil sought to democratize the use of Computers – 20% of all computers used by Brazilian ministries run Linux and other OSS and were expected to increase to 100% within a few months of legislation.
Brazil	Federal Gov't	R&D	Aug. 2004	Govt. signed a cooperative agreement with an OSS company to create a Technology and Knowledge Dissemination Center (CDTC) to promote open standards-based solutions through training and support.
Brazil & South Korea	Interagency	R&D	Nov. 2004	Brazil's National Institute of Technology of the Information (ITI) and the Korean IT Industry Promotion Agency (KIPA) signed an agreement to exchange OSS experiences.
Bulgaria	UNDP	R&D	June 2004	The UN Development Program (UNDP) and Internet Society of Bulgaria (ISOC-Bulgaria) launched a project to help municipal govts. in Southeastern Europe with FOSS.
Cambodia	Executive	Preference	Feb. 2005	Cambodia issued an Open Source Master

				Plan and was expected to come out with an Open Source Action Plan soon after. The Open Source Master Plan lays out a 4-year OSS adoption plan for govt. systems and for development and promotion of OSS and Khmer language functionality.
Canada	CIO Branch	Advisory	Sept. 2003	Canada does not distinguish on the basis of software development models.
China	Beijing Science and Technology Commission	Advisory	August 2002	“Beijing Science and Technology Commission has endorsed Linux as China's most important opportunity to improve its software industry... the commission urged Chinese govt. bodies to consider using Linux with new computer systems, and encouraged private & university software designers to develop Linux and other OSS programs.
China	Ministry of Information Industry	Advisory	Sept. 2002	China's Ministry of Information Industry established an Open Source Alliance to support Linux systems.
China	Ministry of Information Industry	R&D	Aug. 2004	To promote development of China's OSS industry, MII established the Open Source Software Promotion Alliance, which is composed of enterprises, non-profit organizations, representatives from NGOs, & individuals under Chinese govt. guidance.
China, S. Korea & Japan	Multinational	R&D	Sept. 2003	An ongoing collaborative project created to promote OSS replacement of proprietary operating systems. Japan earmarked \$8.6 million for project. At a meeting on April 3, 2004, officials agreed to seek ways of reducing costs of software with Linux. The most recent meeting was in late July 2004 and aimed to promote development and use of OSS.
Costa Rica	Executive	Preference	Feb. 2002	Executive order stated public institutions can use OSS when possible & useful.
Czech Republic	Ministry of Informatics	Advisory	Nov. 2005	Ministry supported the Czech Open Source Software Alliance to provide consultation on OSS projects and assists in representation in the EU's IDABC with regard to OSS.
Denmark	Executive	Advisory	June 2003	The government adopted a “Software Strategy” emphasizing value for money, competition, freedom of choice, and interoperability. Policy doesn't express any preference for open source, but several open source projects were initiated under policy.
Denmark	Ministry of Science and Technology, and Innovation	Advisory	Oct. 2002	Analysis and recommendations drawn up by a working group under the Danish Board of Technology recommended that OSS compete on same level as proprietary software, and for OSS pilot projects.
Denmark	Board of Technology	R&D	Oct. 2002	The Danish Board of Technology released a report stating that the public administration would save 500 million Euros over four years by using open source software. The report also concluded that... open source solutions cannot be dictated as a general principle.”
Denmark	Legislative	Mandatory	July 2007	During a one-year pilot program, government agencies are required to carry Open Document Format (ODF) and

				Microsoft's Open XML format on all computers. The Danish Parliament & a third party will evaluate test program in 2009.
EU	DG XIII, Info Society	Advisory	2003-2004	2003-2004 Work plan encourages use of OS where appropriate for specific program of "Integrating and strengthening the European Research Area."
EU	Directorate Info Society	Advisory	June 2002	eEurope 2005 Action Plan recommended open source for a EU "interoperability framework."
EU	EU Commission	Advisory	Sept. 2003	A Communication from the Commission on the role of eGovernment supporting exchanging experiences in use of open standards and open source amongst public administrations to promote efficiency, productivity, and quality of their services.
EU	EU Telecomm Ministry	Advisory	Dec. 2001	Resolution on network and information security approved by the Council of EU Telecommunications Ministers encouraged the EU Member States to initiate effective and interoperable security solutions based on recognized standards including open source software in their e-government and e-procurement activities.
EU	Ministerial Resolution on E-Government	Advisory	Nov. 2001	The Ministers of Public Service and Administration of the EU Member States, the EFTA Member States, and the accession countries included open source language in a declaration underlining the importance of e-government to the development of Europe's Information Society.
EU	Information Society Technologies Advisory Group	Advisory	Sept. 2002	ISTAG published a report on 'Software Technologies, Embedded Systems, and Distributed Systems' calling for the use of open source licensing for software generated in the Information Society Technologies (IST) program. Under the IST program, the EU Commission launched the 'Three Roses Initiative' to provide funding for the use of open source software in e-government services and e-business solutions in EU Member States.
EU	Information Society Technologies Advisory Group	Advisory	2004	Open, secure, interoperable e-government platforms, applications and multi-modal Services called for use as much as possible open source software solutions for all aspects of inter- and intra-government operations including electronic democracy systems, interaction with citizens and businesses, governmental process re-engineering and knowledge management.
EU	EU Parliament	Advisory	Sept. 2001	Adopted a resolution proposed by the committee on the Echelon Interception System. The resolution urged the Commission and Member States to promote European encryption software and support projects aimed at developing open source encryption software.
EU	Interoperable Delivery of European eGovernment	Advisory	Jan. 2004	The Open Source Observatory sought to provide support for initiatives encouraging OSS uptake & spread OSS good practices.

	Services Program			
	Enterprise D-G Directorate	R&D	Aug. 2003	Call for Tender to establish a service in support of public administrations using OSS with the intention of encouraging the spread of good practice.
EU	Research Institutions	R&D	Dec. 2004/ May 2005	EU provided 2.2 million euros to study OSS in December 2004 and an additional 660,000 euros in May 2005.
EU	EU Commission	R&D	Sept. 2002	Under the Information Society Technologies program, the EU Commission launched the "Three Roses Initiative" to provide funding for the use of OSS in e-govt. services and e-business solutions in EU Member States.
EU	EU Commission IDA	R&D	Nov. 2003	The IDA Open Source Migration Guidelines provided detailed recommendations on how to migrate to Open Source Software (OSS)-based office applications, calendaring, e-mail and other standard applications.
EU	Information Society DG	R&D	May 2003	About 20 projects supporting and developing OSS exist that contribute to development of essential components of a free software infrastructure, and associated development tools or applications.
Finland	Ministry of Finance	Advisory	Oct. 2003	A ministry working paper called for govt. agencies to consider OSS alternatives.
Finland	Joint Venture	R&D	Sept. 2003	The Applied Linux Institute run by the Dept. of Communications and the Institution of Adult Education of Vantaa at the University of Helsinki, & the Dept. of Schooling and Education of the City of Vantaa, (all public institutions), conducted research and development on OS applications.
France	Ministerial	Preference	2003	Ministries of Defense, Culture, and Economy use FOSS operating systems.
France	Ministry of Defence	R&D	Sept. 2004	Ministry formed consortium to develop highly secure Linux-based operating System.
France	Executive	R&D	Aug. 2003	The government launched an open-source content management system to standardize government websites.
France	Commissariat General du Plan	R&D	Oct. 2002	Working group composed of experts from companies and administrative agencies issued a report analyzing the French software industry and examining how the government can best support the industry. Report recommended that public agencies promote the development of free software platforms and open standards.
France	Ministerial	R&D	Nov. 2001	Agency for the Development of the Electronic Administration (ADEA), formerly the Agency for Technologies of Information and Communication in Administration (ATICA), assigned to select open standards to be enforced all over public administrations in order to guarantee full interoperability.
France & China	Interagency	R&D	Oct. 2004	French Atomic Energy Commission and Chinese Ministry of Science and Technology to collaborate to develop Linux-based

				software.
Germany	Bundestag	Advisory	Nov. 2001	Resolution on “Germany’s Economy in the Information Society” promoted FOSS in the federal administration to save on costs.
Germany	Federal Court of Auditors	Advisory	Oct. 2001	The Federal Court of Auditors in a report to the Ministry of Interior adopted position that OSS offered functionalities comparable to commercial software and recommended the use of OSS in the federal administration. The Court of Auditors estimated the use of open source software would yield savings of 100 million Euros. The Bavarian Court of Auditors also announced its support for a transition to open source.
Germany	Bundestag	Advisory	June 2002	Adopted a resolution on ‘Creating an Information Society for All.’ The resolution called for the increased use of open source software in the federal administration and stated that open source is important instrument that can provide for secure and stable IT solutions. The resolution was proposed by the Social Democratic Party, the main party in the governing coalition.
Germany	The Ministry of Economy	Advisory	May 2003	The Ministry of Economy (Federal) announced a decision to stop the systematic promotion of open source software projects. The Ministry adopted a policy to strengthen the competition between free and proprietary software in general. The Ministry will support the new policy objective through neutral public procurement tenders. The Ministry announced that an independent, non-governmental body of experts will formulate criteria for public procurement tenders.
Germany	Ministry of the Interior	Preference	June 2002	Government contracts with Open Source providers for government agencies wishing to make OSS procurements.
Germany	Ministry of the Interior - KBSSt	Preference	July 2003	Published a software Migration Guide which included replacement/migration/integration guidelines for Open Source Software.
Hong Kong	Commerce, Industry, and Technology Bureau	Advisory	March 2003	The Commerce, Industry, and Technology Bureau published a paper entitled “2004 Digital 21 Strategy,” which stated the government will promote OSS within the govt. when viable, and will encourage its use in the private sector with funding.
Hong Kong	Secretary for Commerce, Industry, and Technology	Advisory	Nov. 2002	The Information Technology Services Department (ITSD) issued a circular to all departments, urging them to consider different types of software including open source in procurement and, based on the principle of cost effectiveness, select among the products which meet the basic functional requirements and offer best value for money.
Iceland	Ministry of Justice	Preference	2003	Open source office suite used in all police stations (700 PCs).
India	N.A	Advisory	May 2005	Gov’t distributed millions of free CDs with Tamil and Hindi language OSS.
India	Department of Information	R&D	Sept. 2004	The National Informatics Center created a web site to share the government’s

	Technology			experience in using OSS.
India	Education Ministry	Preference	March 2003	U.S. company donated \$57 million worth of its open source office suite to the Education Ministry.
Iran	High Informatics Council	R&D	Sept. 2004	Government developed OSS alternatives in preparation for a migration for national security reasons and to increase its chances of entry into the WTO through better enforcement of IP laws.
Ireland	N/A	R&D	Dec. 2003	Funded development of a "Knowledge Asset Development System" for Ireland, Europe, & the Middle East.
Israel	Department of Commerce	Preference	Dec. 2003	With its Microsoft contract running out, the Department announced plans to switch most desktops to Open Source Software. Other agencies are showing an interest as well.
Israel	Ministry of Industry, Trade and Labor	R&D	April 2005	Ministry paired with IBM to encourage the use and development of OSS. As part of the plan, the Ministry offers grants of up to \$100,000 for Israeli start-ups.
Italy	National Agency for ICT in Public Administration	Advisory	Oct. 2004	The Plan for Information Society for 2005-2007 called for extensive use of open source applications where possible.
Italy	Ministry for Innovation	Advisory	June 2002	The Minister for Innovation presented a set of Government Guidelines for 2002-2005 to promote technological development. The document called for the adoption of OSS by public administrations...The guidelines also recommended that the govt. launch a national research program on open source.
Italy	Council of Ministers	Advisory	2001	The Council of Ministers endorsed a recommendation by the Senate that urged the administration 'to draft regulations for the examination of open source projects and for the progressive adoption of non-proprietary operating systems and applications by public administrations. Neither the recommendation nor the endorsement was binding on the Berlusconi government that came to power in June 2001.
Italy	Ministry for Innovation	Preference	Feb. 2004	A directive from the ministry stating that in the acquisition of software, the Public Administration must consider OSS and judge software according to transferability, interoperability, dependency on supplier, and the availability of the source code for inspection.
Italy	N/A	R&D	Jan. 2006	The Italian Code of Digital Administration, which became effective on January 1, 2006, required that any software developed by one PA must be made available at no cost, with complete source code and documentation, to any other PA that can adapt it to own needs.
Japan	Ministry of Economy Trade and Industry (METI)	R&D	Feb. 2003	METI planned on spending 1 billion yen in FY04 on OSS Development and Deployment. Procurement policy was open to any new technology and company. METI also promoted OSS collaboration with other Asian countries.
Malaysia	Ministerial	Preference	Aug. 2004	All Government procurements now have a

				strong preference for OSS under the Malaysian Public Sector Open Source Software Master Plan.
Malaysia	Ministerial	R&D	Sept. 2003	The government set up an OSS Competency Center.
Malaysia	Legislative	Advisory	July 2007	The Government of Malaysia decided to encourage use of OSS in the Malaysian Public Sector. Malaysian Administration Modernization and Management Planning Unit (MAMPU) of the Prime Minister Department were responsible for implementing this OSS Initiative.
Netherlands	Ministries of Interior and Economic Affairs	R&D	Jan. 2007	In beginning of 2003, the Dutch government started a program called Open Standards and Open Source Software (OSOSS) to stimulate Dutch government agencies to use open standards in their software and to inform them about open source software. In Dec. 2007, 10 major Dutch cities signed a Manifesto of the Open Cities, signaling that the OSOSS program was working.
Norway	Norwegian Board of Technology	Advisory	Dec. 2004	An independent government advisory board recommended the public stimulation of OSS development through pilot programs.
Norway	Ministry of Labor and Gov't Admin	Preference	July 2002	Norway cancelled contract with proprietary vender to allow for more competition from Open Source and other software.
Norway	Directorate on Public Management	R&D	Aug. 2001	Statskonsult, a state-owned company, carried out a report recommending OSS in the public sector and in education.
OECD	WPISP	Advisory	Oct. 2002	The OECD Working Party on Information Society and Privacy (WPISP) prepared a draft implementation plan of the guidelines for the security of information systems and networks. The draft recommended the utilization of open source technology.
Pakistan	IT Ministry	R&D	Apr. 2004	The Ministry decided to launch an Rs37 million project to train 4,000 government officials from different ministries and departments on the use of open source.
Pakistan	Ministry of Science & Technology	R&D	2003	The govt. established a Task Force for Linux to set up "future directions" for Pakistani IT.
Paraguay	National Science and Technology Council	R&D	May 2005	Paraguay's government studying whether to implement open source in all its entities.
Peru	Legislative	Mandatory	Nov. 2003/ Sept. 2005	Bill required all "Executive, Legislative, and Judicial branches of government, regional & local state entities, and national educational centers to use OSS. A similar bill was introduced in September 2003 by a Vice-President of the Congress. Soft preference bill introduced in June 2003. In September 2005, parliament approved bill that prohibited any public institution from buying systems that tie users into any particular type of software that limited 'information autonomy.
Philippines	Department of Science and	Preference	2001	The Department's Advanced Science and Technology Institute promoted and provided

	Technology			OSS (Bayanihan Linux) in govt. & schools.
Portugal	Council of Ministers	Advisory	Jan. 2002	The Council of Ministers adopted a non-binding resolution promoting use of OSS in the public administration.
Portugal	Ministry of Education	Preference	Mar. 2004	Ministry entered into a 5-year deal with OSS company for secondary schools.
South Korea	IT Industry Promotion Agency	Preference	Feb. 2004	About 1,000 information systems development projects for local autonomous governments would switch to OSS as part of a larger plan to move more systems to OSS.
South Korea	Ministry of Information and Communication	Preference	Mar. 2005	The Ministry promoted OSS use in government by offering a total of 3 billion Won to agencies switching from proprietary software.
South Korea	Ministry of Education	Preference	Oct. 2006	The government launched the National Education Information System (NEIS), building almost entirely on an open source model that used Linux and Sun Microsystems' Solaris.
Singapore	Economic Development Board	Preference	2003	Offered tax breaks to companies that use GNU/Linux operating systems instead of proprietary ones to encourage development of the local software sector.
Slovakia	N/A	Advisory	Aug. 2006	The Slovak Open Source Initiative (SKOSI) founded to create and support free and open source software (FOSS) and free multi-platform infrastructure solutions in the Slovak Republic, as well as to support FOSS integration into education and govt. sectors.
Slovenia	Ministry of Information Society	Advisory	Oct. 2003	OSS and proprietary options given equal consideration in procurements, though the govt. plans to contribute to propagating information and knowledge...of the use of OSS and solutions.
South Africa	Government Information Officers' Council	Preference	June 2003	The Government Information Officers' Council (GITOC) has concluded that: "As OSS offered significant indirect advantages, opting for OSS will be preferable where the direct advantages and disadvantages of OSS and PS are equally strong...open standards will be a prerequisite for all software development, thus contributing to the ease with which OSS can be implemented and adapted; Government encouraged partnerships to foster OSS use. Approved by Cabinet in June 2003.
South Africa	Department of Science and Technology	R&D	Dec. 2003	Department of Science and Technology funded the Open Source Center to promote govt. and educational use of OSS.
South Africa	Government IT Officer's Council	R&D	Aug. 2006	Council investigated use of FOSS in 2003 & made recommendations promoting FOSS applications when proprietary ones offered no advantage. In 2005, the revised policy stated that the South African Government will implement FOSS unless proprietary software demonstrated to be significantly superior. Whenever FOSS is not implemented, reasons had to be provided in to justify the implementation of proprietary software.
Spain	Administración	Advisory	June 2003	The Superior Information Council, which is

	General del Estado			tasked by Spain's General Administration for the approval and diffusion of IT criteria and normalization, recommends adopting OSS when available and when it is satisfactory for the task.
Spain	N/A	R&D	March 2005	The government created the National Center for Open Source Software.
Spain	Development of the Information Society	R&D	May 2006	Government will provide 12 Million Euros for OSS research projects.
Spain	N/A	R&D	May 2006	The National Plan for Scientific Research, Development and Technological Innovation (2004-2007) included a specific budget line for OSS projects, representing 5% of the total budget for R&D for Information Society technologies.
Spain	Legislative	Preference	Jan. 2007	Nearly unanimous resolution in the Parliament promotes the use of OSS in public administration.
Sweden	Agency for Public Management	R&D	Aug. 2003	The Swedish Agency for Public Management (Statskontoret) completed a study to describe free and open source software and to offer suggestions for further work and measures, which recommended that OSS be judged equally with proprietary software in a procurement process.
Sweden	Association of Local Authorities and Regions	R&D	Nov. 2005	The Swedish Association of Local Authorities and Regions launched "Programverket," a project to help public sector adopt or convert to OSS. Programverket would also provide support and facilitate collaboration with OSS in the public sector.
Switzerland	IT Council	Advisory	March 2004	Four-year strategy allows central and local governments to consider OSS alongside proprietary software and sets up "an environment for successful OSS implementation."
Taiwan	Ministry of Economic Affairs	Preference	Oct. 2003	Taiwan will spend US\$3.4 million into promoting OSS development. The government aimed to have 30% of servers and 5% of personal computers operating on open-source software by 2007.
Taiwan	Commission of the Legislative Yuan	Preference	June 2002	The Government sought to encourage R&D and use of OSS. The initiative, which aimed to decrease licensing fees for govt.'s 1.23 million PC's, resulted in Microsoft price cuts for Taiwan.
Taiwan	Government Procurement Agency	Mandatory	June 2006	All government PC's required to be Linux compatible.
Tanzania	Executive	Advisory	Feb. 2003	A National ICT Policy document recommends the use of OSS.
Thailand	Ministerial	Advisory	June 2003	Agreement between the ICT Ministry and the Ministry of Science and Technology to develop and promote OSS in private sector.
Thailand	ICT Ministry	Advisory	May/ Nov. 2003	Agreement with the Thai Software Industry (ATSI) to stimulate OSS development and to distribute one million Linux based computers by May 2004. By August 2003, the government had sold 300,000 PCs.

Thailand	National Electronic and Computer Technology Centre	Advisory	May 2005	The NECTEC director encouraged the use of OSS and said the Thai government has no plans to completely eliminate proprietary software, but would be happy with a 50% OSS penetration rate.
Thailand	Software Industry Promotion Agency	Advisory	Feb. 2005	SIPA driving Linux adoption in government agencies, schools, and universities.
Thailand	National Electronic and Computer Technology Centre	R&D	Sept. 2004	In October 2001, Government officials announced agencies would begin backing initiatives aimed at using the Thai language OSS (Pladoa) in an effort to reduce costs, reduce software piracy, and increase the self-sufficiency of Thailand's economy. Some members of Parliament have also proposed incorporating open source specifications into government IT procurement. Thailand's [NECTEC] is actively involved in the development of OSS office suites and Linux based operating systems. In 2003, NECTEC developed a Linux distribution for schools and government desktops. In January 2004, as a partnership with the Lab School Project, NECTEC began developing Linux server distribution for 921 schools.
United Kingdom	OGC/ e-Government Unit	Advisory	Oct. 2004	The updated version of Government policy on the use of Open Source Software within the UK government specifies software choices should be made on a money-for-value basis, giving no preference to OSS. The National Technical Authority for Information Assurance (CESG) will examine issues on OSS use in govt. systems.
United Kingdom	OGC	R&D	Sept. 2003	November 2002 Case Study and September 2003 "Proof of Concept" Final Report stated that OSS is a "viable and credible alternative" to proprietary software and recommended the public sector consider benefits of development and migration.
United Kingdom	OGC/ e-Government Unit	R&D	Feb. 2003	Nine government agencies evaluated OSS effectiveness and cost-benefits of IT systems based on OSS products. OGC found that OSS was a viable and credible alternative to proprietary software for many applications, but limitations hindering its use and recommended a gradual introduction of OSS as applications improve.
United Kingdom	e-Envoy Office/ Dept. of Industry and Trade	R&D	Feb. 2003	The e-Envoy Office and the Department of Industry and Trade (DTI) adopted interim conclusions on government-funded R&D software outputs...[that] state that if no exploitation route is specified for government-funded R&D software outputs, the default position of the government should be 'to adopt an open source software license which complies with the OSI definition (which includes the GPL and Berkeley style licenses) or a UK-specific analogue of it' [and] 'all government-funded software should be accompanied by appropriate documentation

				that will assist exploitation via OSS license'.
United Kingdom	Office of the Deputy Prime Minister	R&D	June 2005	The government sponsored research at the National Computing Centre in OS applications in the public sector.
United Nations	UNDP	Advisory	April 2003	The UNDP actively promotes government open source software adoption. The Asia-Pacific Development Information Programme (APDIP) of the UNDP launched the International Open Source Network to aid countries in sharing information on OSS. UNDP implemented the DOT Force action items on software development by promoting use and dissemination of open source software within developing countries.
United Nations	UNCTAD	Advisory	Nov. 2003	Called on poor countries to adopt OSS to bridge the digital divide by lowering costs, increasing security, stimulating local economies, and avoiding proprietary lock-in as reasons for adopting OSS.
United States	DoD	Advisory	June 2003	Established rules for open source use within the DoD.
United States	OMB	Advisory	July 2004	Required agencies' procurements must consider cost of ownership and maintenance, as well as risks, security, and privacy of data. Policies are now required to be neutral to both technology and vendor.
Venezuela	Executive	Advisory	August 2002	The government policy articulated open source to be used whenever possible & that proprietary software be used only when necessary.
Venezuela	Executive	Mandatory	Dec. 2004	The decree required all public administration systems to shift to OSS, and in the cases where OSS cannot be used, the agency in need must take requests to adopt other solutions to the Ministry of Science and Technology. The decree also talks about R&D, cooperation, and education in OSS.
Venezuela	Executive	R&D	Nov. 2003	The Venezuelan Academy of Open Source Software opened in Mérida.
Vietnam	Executive	Preference	Mar. 2004	OSS plan for 2004-2008 approved by Prime Minister to develop and accelerate use of OSS for ICT applications; the measure takes steps to encourage OSS adoption in state-owned companies and ministries, but does not require it.
Vietnam	Ministry of Science, Technology, and Environment	R&D	Mar. 2004	The Ministry stated it would spend \$20 million over four years to promote OSS use, develop new OSS applications, and build a skill pool.
Vietnam	Ninth Party National Congress	R&D	Aug. 2002	The Ninth Party National Congress approved 'The Master Plan for IT Use and Development in Vietnam for the Period 2001-2005,' overseen by the Ministry of Science, Technology, and Environment (MOSTE) called for the accelerated development of systems and application software in Vietnamese based Linux or other open source operating systems.

Source: James A. Lewis, "Government Open Source Policies," Center for Strategic and International Studies, August 2007

Appendix H: Taxonomy of Costs

Software:

- Purchase price
- Upgrades and additions
- Intellectual property / licensing fees

Hardware:

- Purchase price
- Upgrades and additions

Support Costs:

Internal

- Installations and set-up
- Maintenance
- Troubleshooting
- Support tools (e.g. books, publications)

External

- Installation and set up
- Maintenance
- Trouble shooting

Staffing Costs

- Project management
- Systems engineering / development
- Systems administration / Vendor management
- Other Administration
- Training

De-installation and Disposal

Indirect Costs

Support Costs

- Peer support
- Casual learning
- Formal training
- Application development
- Futz factor (i.e. labor expense when an employee exploits computing assets for personal use)

Downtime

Appendix I: About Network Centric Warfare

Network Centric Warfare (NCW)

Network Centric Warfare, a concept that emerged from the First Gulf War in 1991, has been refined over the last two decades, but its' core principles have remained the same: an emphasis on modular utility and the ability to actively manage information across a multitude of linked “nodes”, all part of a broader command, control, communications and intelligence network. These networks would connect individual units, soldiers, tanks, planes or other platforms not only to the central command and control nodes, but also directly to one another. Such connectivity would allow for increased sharing of information and intelligence, which will, in turn, lead to dramatically enhanced situational awareness; and this ultimately will lead to enhanced operational effectiveness. In order for militaries in the modern security and operating environment to fully exploit networks and concepts of network centric warfare, the networks themselves must be the following (explained in further detail within Appendix B):

- ***Agile and flexible***
- ***Robust and secure***
- ***Interoperable***
- ***Capable of delivering knowledge to the warfighter rather than solely information***

Given these persistent network C4ISR requirements, the role of software has become increasingly important. Operational necessity demands unified command, control, communications and information sharing across many different systems and platforms, and many modern and modernizing militaries are currently struggling with understanding the best software approach to providing the modularity and interoperability that network centric warfare requires.

Jane's Strategic Advisory Services

Corporate Services Practice
Tate Nurkin
110 N. Royal Street, Suite 200
Alexandria VA, 22314
+ 1 (703) 236 2417