

Legal Issues for the Use of Free and Open Source Software in Government

Nic Suzor

BInfTech/LLB Graduand (QUT)

Professor Brian Fitzgerald

Head of the School of Law

Queensland University of Technology, Australia

Graham Bassett

Barrister, Bank of New South Wales Chambers, Brisbane, Australia

Software licensing has two approaches - proprietary and non-proprietary.

Proprietary methods involve employing a team of programmers and tying them to a non-disclosure agreement. Cloistered for a period of time, they create, test and debug their code. Most importantly, copyright is claimed over the resulting code. Software is marketed as a copyright license and defined as "any product we make available for license for a fee".¹ Bill Gates has made it clear that code is zealously guarded and presented in executable form only: "...a competitor who is free to review Microsoft's source code ... will see the architecture, data structures, algorithms and other key aspects of the relevant Microsoft product. That will make it much easier to copy Microsoft's innovations, which is why commercial software vendors generally do not provide source code to rivals".²

¹ Microsoft Open License Agreement v 6.0, 1 October 2001, para [1] - applicable from 1 July 2002.

² State of New York v Microsoft Corporation, Direct Testimony of Bill Gates, 18 April 2002
[<http://www.microsoft.com/presspass/trial/mswitness/2002/billgates/billgates.asp>],

Typically, proprietary licenses are sold under a Volume License Product Key (VPK) and the consumer is held liable for any unauthorized use of this key.³ A customer can run the program which is defined as the capacity to copy, install, use, access or display the product for the number of copies authorised. A proprietary licensee may not "reverse engineer, decompile, or disassemble products except to the extent expressly permitted by applicable law".⁴ This is contrary to the view that software diversity is best facilitated by reverse engineering.⁵ A licensee may not rent, lease, lend or host products.⁶ In return, the user is offered a limited warranty that the product will "perform substantially in accordance with our user documentation" for a period up to ninety (90) days from first running the program.⁷ Licensees are dependent on the vendor for upgrades and patches. Traditionally upgrades enabled a licensee to purchase modifications when, and as, they saw fit. Microsoft's Software Assurance scheme requires a user to buy an upgrade subscription as part of the license of a product.⁸ Critics claim that this upgrade scheme applies a fee to the licensee even if no upgrade is provided in that period and this merely offers a "right to upgrade that previously existed without any requirement for advanced payment to preserve the right".⁹

Non-proprietary, free software, on the other hand, is software that provides users with freedom in the way in which they can deal with the software. This freedom, at a minimum, is the freedom to "run, copy, distribute, study, change and improve the software". Richard Stallman, of the Free Software Foundation, articulates the four kinds of freedom guaranteed by free software:¹⁰

- The freedom to run the program, for any purpose.
- The freedom to study how the program works, and adapt it to your needs.
- The freedom to redistribute copies so you can help your neighbor.
- The freedom to improve the program, and release your improvements to the public, so that the whole community benefits.

para [307] 20 April 2002.

³ Note 1, para [4].

⁴ Note 1, para [7].

⁵ B. Fitzgerald, C. Cifuentes, A. Fitzgerald and M. Lehmann, "Innovation, Software and Reverse Engineering" (2001) 18 *Santa Clara Computer And High Technology Law Journal* 121; B. Fitzgerald "Intellectual Property Rights in Digital Architecture (including Software): The Question of Digital Diversity?" [2001] *EIPR* 121.

⁶ Note 1.

⁷ Note 1, para [9a].

⁸ Note 1, para [11].

⁹ A NZ company has made a formal complaint about the impact of this new 'software-as-service' paradigm. See "Complaint to the Commerce Commission by Infraserv Limited as to certain anti-competitive behaviour of Microsoft NZ Limited", [www.clendons.co.nz], 9 May 2002.

¹⁰ Free Software Foundation, *The Free Software Definition*, <<http://www.fsf.org/philosophy/free-sw.html>>, at 8 December 2003.

Free software is often developed by hundreds or thousands of individual developers, many working in their spare time, continuously and contemporaneously, all around the globe. Code is continuously being scrutinised by interested parties. If a flaw or bug is discovered, a bug-fix can be implemented and disseminated almost immediately. Similarly, if a better solution is developed, the old solution can be easily replaced. Development and testing is quick, and improvements are readily available. It is for these reasons that open source software is often of higher quality than its proprietary counterpart.

Benefits of open source for governments

The first benefit many people see in open source software is that it may allow a lower total cost of ownership for any given solution. Free software does not require a product key for every computer, or user, or site. As Moglen points out, the provider of a “fully redistributable system containing only free software can reduce the unit cost of software to zero”,¹¹ leaving the customer to pay only for installation and support.

The development model of free software is also more efficient than that of proprietary software – a free software developer can re-use code that was written for any other project, by any other developer, rather than having to re-invent the wheel each time a particular solution is required.

NOIE, the National Office for the Information Economy, states that the Federal Government considers that “the main issue is to determine the cost, benefits and risks of using either open source or proprietary software in a given situation. The Government will encourage trials of open source software within the framework of fit-for-purpose and value-for-money”.¹²

But cost should not be the only factor when governments evaluate software solutions. The emphasis of the free software movement is on freedom, not cost. Many advocates argue that open standards are crucial in any government acquisition of software. Open standards are file formats and communication protocols which are agreed upon by community consensus and are not controlled by proprietary companies. What this means is that when we use open standards, we are always guaranteed to be able to access our data, even in the future when the original software or hardware becomes obsolete. It means that the government is not mandating the use a particular vendor’s systems by communicating only through a proprietary format, nor is it locking future generations into using the same proprietary systems. It means that the standard is always published; we are free to comment, criticise, and modify it and we know exactly what information is being stored. While open standards do not equate exactly to open source, open source software is generally more likely to use open standards because of the public consultation inherent in the development process.

¹¹ Eben Moglen, *Free Software Matters: Free Government, II*, <<http://emoglen.law.columbia.edu/publications/lu-24.html>>, at 7 December 2003.

¹² National Office for the Information Economy, ‘Better Services, Better Government’, November 2002 <http://www.noie.gov.au/publications/NOIE/better_services-better_gov/Better_Services-Better_Gov.pdf>, at 8 December 2003, at 21.

The next benefit of free software lies in the effect that readily available source code and quick development releases has on security. Bruce Schneier argues that security is better served by full disclosure of vulnerabilities and fast releases of patches.¹³ In proprietary software, vulnerabilities are harder to identify because there is no access to the source code. Once a vulnerability is found, only the licensor has the power to patch the hole, and users must wait for an official update. Conversely, free software benefits by having greater public scrutiny of the source code, faster release times, and, if necessary, the problem can always be fixed in-house.

A recent report into the use of Free and Open Source Software in the U.S. Department of Defense identified that free software was vital to information security in three ways:

- the free software community has “produced infrastructure software [...] with low rates of software failure combined with early and rapid closure of security holes, which makes such systems useful as the security linchpins in broader security strategies”;¹⁴
- the communities have had a “long-term fascination with developing more and more sophisticated applications for identifying and analyzing security holes in networks and computers, resulting in [open source] products [...] that are invaluable to in-depth analyses of security risks”;¹⁵
- Free software “contributes to security by making it possible to change and fix security holes quickly in the face of new modes of cyberattack. This ability, which allows rapid response to new or innovative forms of cyberattack, is intrinsic to the [free software] approach and generally impractical in closed source products”.¹⁶

Professor Bill Caeli argues that since software can not be trusted to be secure at present, users, and particularly governments, must be able to examine the workings of the systems to be satisfied of their security, and be able to implement tougher security measures where the system is found lacking. Professor Caeli believes that “open source licensing represents the ideal for the evaluation of the underlying security architecture in the operating system and the allied mechanisms that activate and support necessary hardware security features”,¹⁷ and that “Reasonable prudence would thus suggest movement towards an open source solution”.¹⁸

A final benefit of free software is that it provides a framework to give back the benefits of publicly funded software development to the public. When a government

¹³ Bruce Schneier, ‘Full Disclosure’, *Crypto-Gram* 15 November 2001, <<http://www.schneier.com/crypto-gram-0111.html>>, at 8 December 2003.

¹⁴ Mitre, ‘Use of Free and Open-Source Software (FOSS) in the U.S. Department of Defense’, 2003 <<http://www.egovos.org/pdf/dodfoss.pdf>>, at 8 December 2003, at 20.

¹⁵ *ibid.*

¹⁶ *ibid.*

¹⁷ Prof. William Caeli “Security with Free and Open Source Software” in B Fitzgerald and G Bassett (eds) *Legal Issues Relating to Free and Open Source Software* (2003) at 112.

¹⁸ Note 17 at 113.

develops publicly funded software, there is a strong argument that, subject to issues of security and confidentiality, the software should be made available to the public. While not all internally developed software may be suitable for public release, use of free software may provide a framework to release code to the commons without attracting liability or requiring further expenditure to support the software.

Government development of software

The Commonwealth and States are owners of copyright in original literary works made by, or under the direction and control of, the Commonwealth or the State.¹⁹ As a nation Australians "have agreed to unite in one indissoluble Federal Commonwealth under the Crown".²⁰ The *Copyright Act* deems the Crown to be the copyright owner for works made by or under its control.²¹ The Crown also retains copyright on material first published in Australia under its direction²² and the Crown is bound by the express words of the copyright statute.²³ However, provisions relating to Crown ownership may be modified by agreement so as to place ownership with another author or maker.²⁴

The proprietary software developer reaps a number of benefits from engagement with government on a departmental, contractual level. They benefit from payment of license fees for the adoption of their software. Furthermore, the system established would require users who have had no choice in the matter to adopt the software of this proprietary company in order to be interoperable, thereby reinforcing proprietary market penetration. In any agreement with a government entity, the proprietary organisation might also supply the contractors to develop the system, recouping substantial fees for these contractors. Section 179 of the *Copyright Act* would be invoked to retain copyright over the systems developed, thus gaining further licensing fees from the capacity to on sell this software to other government departments. The Crown would not be indissoluble for such agreements. Source code of any adaptations made under the government contract can be further exploited by the proprietary organisation in other markets as they have no obligations on them to supply source code upon distribution. The government department is beholden to this organisation for any modifications to the software. The privacy implications for proprietary organisations having access to personally identifying data in such government systems is another issue that needs to be considered in a paper of its own.

Open source development of software provides a different scenario. The government could pay external developers but design their contracts so that the government maintains control and direction and owns the resultant product. The GPL is an agreement modifying Crown ownership of copyright. When combined with indivisibility of the Crown, distribution requirements of the GPL mean distribution

¹⁹ *Copyright Act* 1968 Cth, s176(2).

²⁰ *The Constitution* 1900 Cth, Preamble.

²¹ Note 19, s176.

²² Note 19, s177.

²³ *Bropho v. Western Australia* [1990] HCA 24.

²⁴ Note 19, s179.

only takes place when a modified work is distributed beyond government departments, instrumentalities or statutory authorities. The government would retain copyright over its code but be able to distribute this code and modifications to other departments at no extra licence fee and only for the cost of distribution. Departments would share in a common pool of software knowledge as modifications were made to address particular issues. Contractors could be paid to ensure warranties for the operation of the software. Access to the source code would allow departments to rectify bugs more quickly. If the interfaces to government systems were open source, the proprietary companies would need to ensure they are interoperable in order to maintain their market share of users utilising their applications. Moreover, if users could access such government systems with open source applications, the promotion of such applications, most of which are significantly cheaper than proprietary alternatives, would be enhanced. Open Office may become mainstream and the competitor products be made to realise significantly reduced pricing in order to remain competitive.

If the government is to distribute beyond its own entities, those who take the GPL software must take under the same license and provide any modified code to others to whom they distribute. This is one of the criticisms of proprietary owners who argue that government funded code under the GPL can never be commercialised. In the alternative, Pavlicek questions the windfall that private organisations reap in closed-code arrangements with government:

Which is more deplorable: that a few profit-making software companies won't be able to make as much profit from publicly funded software, or that the public who already paid for the software once with their tax dollars will have to pay for it again when the large software company puts it into their closed-source product?²⁵

Where the line will be drawn with regard to use of GPL or proprietary software in government requires a government's assessment of their role in society. Is their role to promote and enhance private enterprise? Proprietary organisations say so:

"The principle role of government and universities in the ecosystem is to encourage private enterprises and individuals capable of developing these innovations commercially."²⁶

Is the role of government to maintain the public good? There is evidence the Crown may be rethinking the strong controls they have over copyright under the statute. An inquiry has been called to examine whether statutory bodies and other "emanations or agencies of government" are independent bodies as far as copyright ownership is concerned.²⁷

²⁵ Russell Pavlicek, "Don't Fear the GPL", *InfoWorld*, August 23 2002, <http://www.infoworld.com/article/02/08/23/020826opsources_1.html> at 9 December 2003.

²⁶ GNU general Public License (GPL), <<http://www.microsoft.com/resources/sharedsource/Articles/GNU.msp>> at 9 December 2003.

²⁷ Copyright Law Review Committee, "Terms of Reference – Crown Copyright", 5 December 2003, para 1(b), <www.clrc.gov.au>, at 8 December 2003.

One observer argues that governments are so entranced by the minimalist role they should play in a free market economy that they have become blind to the benefits of government intervention. He calls on governments to take a more active role:

When government steps aside, it is not as though nothing takes its place. When governments disappear, it is not as if paradise prevails. It is not as if private interests have no interests, as if private interests do not have ends that they will pursue. To push the anti-government button is not to teleport us to Eden. When the interests of governments are gone, other interests take their place. Do we know what those interests are? And are we so certain they are better?²⁸

Types of free software licences

Broadly speaking, there are two main types of free software licences. The simpler licences, for example the BSD and MIT/X11 licences, allow redistribution and use in source and binary forms, with or without modification, on the condition that the copyright notice is retained and that any applicable warranties are disclaimed. There is no requirement that derivatives of the free software be free themselves. On the other hand, the restrictive free software licences, like the GNU General Public Licence (GPL), attempt to create a contributory commons by requiring that any redistribution of the software or its derivatives is released under the free licence.

The obligation to redistribute source code

This obligation to redistribute must be clearly understood by any user of free software. If the government decides to use free software, it must be aware of when an obligation to distribute source will arise. For licences like the GPL, a government will be obliged to distribute the source for any derivative works it makes and distributes.²⁹

Due to uncertainties in the licence, it is not clear exactly when a derivative work will be created. Modifications to the software are clearly derivative works and will be treated as such. The difficulty lies in determining when new programs, which simply make use of free software, or are designed to operate with free software, will be treated as derivative works. Richard Stallman argues that any use of code released under the GPL creates an obligation upon the using software.³⁰ However, because the GPL appears to carve out a set of rights from copyright law, it would appear that the better construction would be that of Larry Rosen, who argues that "The primary indication of whether a new program is a derivative work is whether the source code of the original program was used, modified, translated or otherwise changed in any

²⁸ Lawrence Lessig, 'Commons and Code', Media and Ent. L.J. 405 (Symposium: Key Address)

²⁹ "The General Public License (GPL)", Version 2, June 1991, <<http://www.opensource.org/licenses/gpl-license.html>>, 7 December 2003, cl 2(b).

³⁰ Richard Stallman, 'Why you shouldn't use the Library GPL for your next library', 1999, <<http://www.gnu.org/licenses/why-not-lgpl.html>>, at 7 December 2003.

way to create the new program”,³¹ and further that “The meaning of derivative work will not be broadened to include software created by linking to library programs that were designed and intended to be used as library programs”.³² Accordingly, it is possible to create new software that uses and relies upon free software, without creating a derivative work.

The distinction, though fine, is important. If a program is a derivative of another work which is licensed under the GPL, any distribution of the new program must also be under the GPL. On the other hand, if the new software is not a derivative, the developer is free to release the software on any terms. For governments, this can be very important to prevent the obligation to release sensitive or confidential information. Software development that will contain or use such information should be carefully planned, before implementation, to sit above any restrictive free software, to safely avoid the obligation to disclose.

Merely creating a derivative work on its own will not give rise to an obligation to publish under a free software licence. The derivative work must be ‘distributed’. Again, what exactly constitutes a ‘distribution’ is not clear. It is apparent that a distribution within one organisation can not be considered a ‘distribution’ under the GPL.³³ Similarly, Eben Moglen, chief counsel for the Free Software Foundation, takes the view that “Federal Government agencies may share free software without making a ‘distribution.’”³⁴ So too, in Australia, sharing of code between government departments would not give rise to an obligation to make the source code available to the public.³⁵ However, where the software is shared to or from a statutory corporation, there will be a stronger argument that a ‘distribution’ has taken place. Where a commercial body exists to fulfil a government process, but is otherwise independent from the government, it is probable that any distribution would not be taken to have been made between two aspects of one indivisible crown; rather, the presumption would arise that a distribution between two separate entities had taken place. So, any software that contains sensitive or confidential information, if it forms a derivative of any restrictive free software, can be shared between government departments without requiring disclosure of the source, but care must be taken to avoid distributing to third parties, including statutory corporations, even under obligations of confidence like Non-Disclosure Agreements (NDAs).³⁶

Finally, on the subject of sensitive or confidential information, it must be made clear that merely using free software to create or store the information will never give rise to an obligation to disclose. The concern only arises when such information is used to

³¹ Larry Rosen, ‘Geek Law: Derivative Works’, <<http://www.linuxjournal.com/article.php?sid=6366>>, at 7 December 2003.

³² *ibid.*

³³ Frequently Asked Questions about the GNU GPL <<http://www.gnu.org/licenses/gpl-faq.html#InternalDistribution>>, at 7 December 2003.

³⁴ Email from Eben Moglen to Prof. Fitzgerald, Wed 3 December 2003.

³⁵ This is in line with the constitutional notion of one indissoluble Federal Commonwealth under the Crown.

³⁶ Frequently Asked Questions about the GNU GPL <<http://www.gnu.org/licenses/gpl-faq.html#DoesTheGPLAllowNDA>>, at 7 December 2003.

create or modify the software itself; an end-user who does not modify source code will never be under such an obligation.

Requirements of the Trade Practices Act

Many free software licences purport to disclaim all warranties, whether express or implied, to avoid the possibility of free software developers being held liable for any fault of the program. In Australia, the *Trade Practices Act* (TPA)³⁷ provides certain non-excludable warranties where a corporation is carrying on a business. The act applies to the Commonwealth Government and Commonwealth agencies, when either is carrying on a business, but only Commonwealth agencies can be fined or prosecuted.³⁸

The Trade Practices Act establishes several important consumer protection measures. Importantly, it prohibits misleading or deceptive conduct³⁹ and the making of false or misleading representations⁴⁰, and implies warranties as to title and of quiet enjoyment⁴¹, conditions that goods (including software) will be fit for the purpose supplied⁴² and of merchantable quality,⁴³ and a condition that goods supplied by reference will correspond with the sample.⁴⁴ These implied conditions and warranties can not be excluded by contract.⁴⁵ These provisions apply when a corporation or government is acting in 'trade or commerce'. Peter James notes that "where software is supplied by way of gift, not sale, this requirement nevertheless would be satisfied if the software supply is part of a commercial dealing or if the supply is connected (even indirectly) with advancing or protecting the commercial interests of the supplier.⁴⁶ This means that the implied conditions will generally only apply to suppliers of free software, and not individual developers. If a government, or a government agency, begins to engage in a commercial or a related supply of software to consumers, it must be aware that these provisions impose certain minimum levels on the quality of any software it provides, as well as to the way the software is represented. On the other hand, if the government developer merely gives code to an open source project outside of a business relationship, no liability should arise.

Due to the generally loose wording of exclusion clauses found in free software licences, exclusion clauses common in free software licences may not be effective at limiting liability for negligence and consequential damages. Peter James notes that

³⁷ *Trade Practices Act* 1974 (Cth).

³⁸ Note 37, s 2A.

³⁹ Note 37, s 52.

⁴⁰ Note 37, s 53.

⁴¹ Note 37, s 69.

⁴² Note 37, s 74B.

⁴³ Note 37, s 74D.

⁴⁴ Note 37, s 72.

⁴⁵ Note 37, s 67.

⁴⁶ Peter James "Open Source Software: An Australian Perspective" in B Fitzgerald and G Bassett (eds) *Legal Issues Relating to Free and Open Source Software* (2003) at p 78, citing *Fasold v Roberts* (1997) 70 FLR 489.

the “courts look at the provision as a whole and, if the exclusion attempts to limit liability for the very purpose of the contract, it will need to be clearly and unambiguously drafted to survive challenge”, which the GPL is not.⁴⁷ For these reasons, anyone supplying software under the GPL and similar licences will have to be aware that they may be liable for damages not only for direct losses, but also for consequential losses, including loss of profits or data.

Enforceability of the GPL

There is considerable debate over the enforceability of the GPL, and whether it is to be construed as a licence or a contract. Specifically, if it is a contract, is there valid consideration to create an enforceable contract? On the other hand, if it is considered to be a copyright licence, is it able to enforce the requirements that users distribute any derivative works under equivalent terms? Ben Giles argues that since the only promise that a free software user makes is to redistribute under the GPL if and only if they *choose* to distribute a derivative work, that promise is not sufficient and there is no consideration to support a valid contract.⁴⁸ This argument rests on the doctrine of illusory consideration, which means that promises that are only to be carried out at the promisor’s discretion can not create a binding contract.⁴⁹ There has been no recent significant interpretation of this doctrine. Arguably, due to significant changes in the way in which parties do business online, the doctrine may have lost some significance or relevance in recent years.

On the other hand, Eben Moglen suggests the GPL is a copyright license, not a contract: “Licenses are not contracts: the work's user is obliged to remain within the bounds of the license not because she voluntarily promised, but because she doesn't have any right to act at all except as the license permits.”⁵⁰

As yet, there has been no significant litigation concerning the enforceability and classification of the GPL. Moglen suggests that “there have been no such controversies because nobody thinks they're going to win them”.⁵¹ Maureen O’Sullivan notes that the threat of damage to a firm’s reputation from the watchful hacker community, as well as the possibility of a lengthy court case, has been

⁴⁷ Note 46 at 80.

⁴⁸ Ben Giles, “Consideration’ and the open source agreement’ (2002) 15 *Computers & Law* 16

⁴⁹ *British Empire Films Pty Ltd v Oxford Theatres Pty Ltd* [1943] VLR 163 per O’Byrne J.

⁵⁰ Eben Moglen, “Free Software Matters: Enforcing the GPL, I”, 12 August 2001, [<http://emoglen.law.columbia.edu/publications/lu-12.html>] 25 January 2002. See also: B Fitzgerald, “Digital Property: The Ultimate Boundary?” (2001) 7 *Roger Williams University Law Review* 237; B Fitzgerald, “Commodifying and Transacting Informational Products Through Contractual Licences: The Challenge for Informational Constitutionalism” in CEF Rickett and GW Austin (eds), *Intellectual Property and the Common Law World*, Oxford, Hart Pub, 2000, 35.

⁵¹ Moody, G (2001), *Rebel Code*, London: Penguin, p.313.

successful over the last decade to ensuring that firms comply with the terms of the GPL.⁵²

It is clear that even though the GPL has not been tested in court, questions about its technical legal enforceability are not barriers to its use – significant compliance with its terms can be expected to continue well into the foreseeable future.

The other concern about free software licences is that generally, a gratuitous licence can be revoked at will.⁵³ This means that, in the case that one single entity controls a significant portion of the copyright in the source code for a free software package, that entity may be able to terminate the licence and users will no longer be entitled to copy or redistribute the software. Jeremy Malcolm calls this “one of the best kept secrets of the open source movement”, and notes the potential danger that could occur if an upstream developer revoked the licence, causing all derived projects to be rendered invalid to the extent that they are derived from the original. Of course, this proposition also goes the other way – in large, distributed projects, it will be hard for any one person to revoke the licence on any part of the project.

However, in the case that a licence is revoked, it is possible that the doctrine of estoppel may prevent the copyright owner from asserting his or her rights. Equitable estoppel has been developed to prevent a person from reneging on a promise (i.e., that another can use, modify and distribute his or her software, perpetually), where that other person has relied on the promise (by using the software), and it would cause loss to the person if the promise were not kept.⁵⁴ Again, getting to this stage in legal proceedings would be quite rare. While revocation may be technically possible, it is unlikely to occur in the face of tough public opposition and a vigilant open source community. Regardless, as has been demonstrated over the last six months with regard to the *SCO v IBM* suit, the hacker community is more than willing to replace any code for which the licence has been revoked or that otherwise infringes copyright. For these reasons, the question about the revocability of free software licences is much more of an academic than a practical concern.

Democrats open source bill

The Australian Democrats aim to legislate consideration of open source software for public agency procurement contracts. An initial attempt to legislate at a state level⁵⁵ in South Australia was recently refined and presented as a Bill to the Federal Parliament. The *Financial Management and Accountability (Anti Restrictive Software Practices) Amendment Bill 2003* aims to redress concerns that “a small number of

⁵² O'Sullivan M, 'Making Copyright Ambidextrous: An Expose of Copyleft', (2002) 3 *The Journal of Information, Law and Technology* <<http://elj.warwick.ac.uk/jilt/02-3/osullivan.html>>, at 7 December 2003.

⁵³ *Trumpet Software Pty Ltd & Anor v OzEmail Pty Ltd & Ors* [1996] 560 FCA 1.

⁵⁴ *Waltons Stores (Interstate) Ltd v Maher* (1988) 164 CLR 387.

⁵⁵ *State Supply (Procurement of Software) Amendment Bill*, <http://www.parliament.sa.gov.au/dbsearch/lcbills_search.asp> at 15 September 2003.

software manufacturers have a disproportionate and restrictive hold on the supply, use and development of software".⁵⁶

The aim is to mandate consideration of open source:

An Agency must, in making a decision about the procurement of computer software for its operations, have regard to the principle that, wherever practicable, an Agency is to use open source software in preference to proprietary software.⁵⁷

A vendor participating in a government software procurement program must ensure its software "follow industry-wide accepted standards that are open to all vendors and display an open format".⁵⁸ The data that is used in such software "will be kept at all times in a format that is completely documented in public".⁵⁹ Where agencies have purchased proprietary software it is incumbent on them in their annual report to list details of such purchases and details as to why any open source alternative was not procured.⁶⁰

How does the Bill define open source software? It does not specifically require that they have a licensing model accepted by the Open Source Initiative. Instead, the definition asserts:

open source software means computer software the subject of a licence granting a person a right:

- (a) without any limitation or restriction, to use the software for any purpose; and
- (b) without any limitation or restriction, to make copies of the software for any purpose; and
- (c) without any limitation or restriction, to access or modify the source code of the software for any purpose; and
- (d) without payment of a royalty or other fee, to distribute copies of:
 - (i) the software (including as a component of an aggregate distribution containing computer software from several different sources); or
 - (ii) a derived or modified form of software (whether in compiled form or in the form of source code), under the same terms as the licence applying to the software.⁶¹

The Initiative for Software Choice (ISC) has opposed the legislation proposed by the Australian Democrats. In responding to the earlier bill proposed in the South Australian Parliament, the group wrote a letter to the Premier, Mike Rann stating:

⁵⁶ *Financial Management and Accountability (Anti Restrictive Software Practices) Amendment Bill 2003*, Preamble

⁵⁷ Note 56, s44A(1)

⁵⁸ Note 56, s44A(2)(a)

⁵⁹ Note 56, s44A(2)(b)

⁶⁰ Note 56, s44A(3)

⁶¹ Note 56, s44A(4)

The ISC strongly supports the development and adoption of all kinds of software – OSS, hybrid and proprietary. All models have a place in the highly competitive software market. Only in this manner, through vibrant and open competition, does the whole of the market thrive, and consumers – both public and private – reap tremendous benefits. Standing in stark contrast to open competition are state-mandated software preferences. These “preference” policies strip merit out of the process by using access to source code as a proxy for ICT project success.⁶²

...

The result would be reduced options for software acquisitions, largely eliminating proprietary offerings that might be the best solutions for the given need.

Additionally, constituents would suffer because the best solutions could never truly be acquired, with at least one development model – proprietary software – being restricted from agency consideration. Further, South Australia’s primarily proprietary-based, ICT industry would be harmed because of foreclosed access to important state market opportunities.”

The ISC group is reported as saying such government mandates would be a barrier to free trade agreements.⁶³

The proposer of the Democrats Bill, Senator Brian Greig, rebutted these claims, specifically referring to groups such as ISC. Senator Greig points out that many current government systems, often unwittingly, mandate use of proprietary systems because software procurement choices have not considered open source alternatives, and will not work with open formats or open source software. Greig argues:

The forces of proprietary software and their supporters have tried to portray this bill as being protectionist in nature, one that tries to pick software favourites. It is in fact the complete opposite. Currently, we have a system that is largely based on proprietary formats, a system that does pick favourites. Removing this and opening up the playing field to all, is the *raison d’être* for this bill.⁶⁴

Senator Greig points out that when the Thai government mandated use of open source software it was able to provide a hardware and software solution around the same price as the cost of licenses for Microsoft products alone on the same machine. The result was that Microsoft dramatically reduced its prices in order to stay competitive in the government contract area. Greig claims that Microsoft would recoup lost revenue when they provide upgrades. The key was to obtain, and then be able to control, the contract. “Microsoft’s actions echo the words of Henry Ford

⁶² Letter from The Initiative for Software Choice to The Honourable Mike Rann, 10 June 2003 <http://softwarechoice.org/download_files/DearSouthAustraliaRann.pdf> at 22 September 2003.

⁶³ Simon Hayes and James Riley, *The Australian IT Today*, “Open Source Trade Clash” July 1, 2003.

⁶⁴ Senator Brian Greig, Second Reading Speech.

when he offered to give away his cars provided he could keep the monopoly on spare parts. It is this type of monopoly that the use of proprietary formats maintains.”⁶⁵

The Australian Capital Territory, on 10 December 2003, became the first Australian jurisdiction to pass the Democrats’ Bill. The bill was amended to require ACT entities to consider open source software and avoid “software that does not comply with open standards”⁶⁶ and software “for which support or maintenance is provided only by an entity that has the right to exercise exclusive control over its sale or distribution”⁶⁷. The amended bill also ties the definition of open source software to that of the Open Source Initiative⁶⁸, and adds a three year sunset clause to the enactment.⁶⁹

Conclusion

There are many significant advantages to a broad government adoption of free software, ranging from potential cost savings, adoption of open standards and protocols, wider use of stronger, more flexible and more secure software, to the social benefit of promoting a contributory commons of free software. However, governments have to be aware of the obligations that may be imposed by the use and redistribution of open source software, and when exactly these obligations will arise. Governments must also be aware of the effect that implied warranties may have upon the sale or supply of free software by virtue of the *Trade Practices Act*, and the limitations inherent in the exclusion of liability clauses in many free software licences.

The evaluation of whether a government should use free software for any given application is a complex matter. However, with the continual increase in quality and quantity of available solutions, coupled with increased understanding of the advantages and obligations involved, we can expect and hope to see much more widespread use of free software in governments in the near future.

⁶⁵ *ibid.*

⁶⁶ *Government Procurement (Principles) Guideline Amendment Act 2003 (ACT)* (Inserts s 6A into *Government Procurement (Principles) Guideline Act 2002 (ACT)*), s 6A(1)(b)(i).

⁶⁷ Note 66, s 6A(1)(b)(ii).

⁶⁸ Note 66, s 6A(4).

⁶⁹ Note 66, s 6A(5).