2006

FLOSS as Democratic Principle

Mark Perry
University of Western Ontario, mperry@uwo.ca

Brian Fitzgerald
Queensland University of Technology, bf.fitzgerald@qut.edu.au

FLOSS as Democratic Principle

Professor Mark Perry[**] and Professor Brian Fitzgerald[*]

Contents

** Faculty of Law, Faculty of Science, University of Western Ontario, written whilst Visiting Fellow at Queensland University of Technology 2005-2006.

* School of Law, Queensland University of Technology, Brisbane, Australia.

A. Introduction: Free Software[1] as Democratic Principle

Free software – software in which the human readable source code is disclosed and distributed to the world - provides an excellent opportunity to "scrutinise" how software operates. By way of contrast, software that is distributed in binary form with the source code not disclosed (closed) promotes secrecy and ignorance as to how the software operates. If I were just about to be convicted of murder or to elect a President through a democratic process that relied on software I would feel more comfortable knowing the *inner thoughts* of that software – for software as a discursive practice has the power through its coding by humans (normally large private corporations) to construct knowledge much the same was as language, and embodies the thoughts of those that build it.

Our thesis is that core software infrastructure in a vibrant democracy must be able to be scrutinised, reviewed and made accountable by any citizen through access to the source code. At present, free software provides that opportunity. What is more, free software allows citizens to better participate in and improve upon the process of democracy.

In this paper we examine this new justification for the use of free software in the public sector or government, which we label "free software as democratic principle." There is growing interest in and rhetoric about the ability of free software to bring more transparency to core software infrastructure within a governmental system. Our argument is that free software should be deployed in core democratic infrastructure because it will provide the level of transparency and openness that is required for the effective functioning of democratic processes.[2] Some free software development could be seen as an intellectual infrastructure and should then be fostered for maximum public benefit.[3] For example, core software infrastructure for voting process or electronic court processes should be transparent, i.e. available to be monitored and understood by any member of a democratic community. It would be a sad day for the functioning of a democratic system if inherent and/or coded bias in a software program skewed the result of an election or the determination of innocence or guilt of a person in court The purpose of this article is to outline an argument of the notion of "free software as democratic principle"; meaning free software should be deployed in core democratic infrastructure to sponsor accountability and transparency, and ultimately access to knowledge.

---

1 Throughout this article we use the term Free Software and Open Source interchangeably, unless referencing specific projects where other terms are used. There are important divisions within the 'free' and 'open' source communities, but these are not the focus of this study where the focus is the accessibility of the code.

2 On this notion consider: B Frischmann, "An Economic Theory of Infrastructure and Commons Management" Review Volume 89 No. 4 (2005) p917-1030.

3 *Idem*.

B. Background – Key Concepts

i) "Software as Discourse"

The starting point for this discussion must be the ability of software to construct meaning. We believe that software is a form of discourse – meaning that it allows things to be seen – and that the shape, form or coding of the software is vital to the way we see the world.[4] In fact we would go so far as to say that software like discourse (e.g. language) has a tremendous capacity to structure knowledge and meaning within any given community. This should not be such a revelation for after all "programming" is about telling computers how to process information.

From this starting point we would argue that there is potential for software employed in core democratic infrastructure to be coded in a manner that serves the interests of one particular group, person or party. Democratic principle demands that we have safeguards against this happening. One way to implement a safeguard in this type of situation and to instil confidence in the software across the general community is to allow the source code to be disclosed.

ii) "E-Governance"

Governments are increasingly dependant on using software for effective management of many services that are core to the administration of the state. Whether this is for voting support systems (whether directly in poll booths, for distance voting, or vote counting), the management of the justice system, or execution of the tax system. The term 'e-governance' has become a theme for the promotion-improved performance in service delivery and improved participation of the citizen in the public sector.

The adoption and development of Free Software continues to grow throughout the world no more so than in the area of government. Local and central governments in countries such as Australia, Brazil Germany, India, Peru, Switzerland, Thailand, and Ukrane, have been extolling the virtues of Free Software adoption for the public sector.[5] To this point reasons for adopting Free Software solutions have focussed on cost, security, error detection, the promotion of open standards and community benefit.

---

4 See further B Fitzgerald "Software as Discourse: The Power of Intellectual Property in Digital Architecture" (2000) 18 Cardozo Journal of Arts and Entertainment Law Journal 337 reprinted in P Yu, (ed.) Market Place of Ideas: 20 Years of Cardozo Arts and Entertainment Law Journal (Kluwer 2003).

5 A preliminary study by The Center for Strategic and International Studies chart of national approaches to the adoption of free software is to be found at, updated December 13, 2004 "Government Open Source Policies" that shows 45 nations have a central government policy approved or considered for adopting Free and Open Source Software.

iii) Participation in and Access to Knowledge in the E-Democracy

The great challenge of the digital environment is to find ways to be able to utilise the great advances technology has allowed in terms of accessing knowledge. In relation to digital content and entertainment the desire is for seamless and lawful access to sharing of knowledge in a way that promotes knowledge, culture and economy. In terms of the intellectual and democratic intent of a community there is also a real question as to how we can better implement democracy through technology by sponsoring greater participation in and access to knowledge.

iv) Free Software – The Essentials

A grass roots movement started by free software guru Richard Stallman in the 1980s has revolutionised the way we think about software development and distribution. Stallman was frustrated with the fact that he could not access the source code (the human readable code) of software that was controlling a Xerox printer in his lab at MIT. His quest for opening up access to source code in software has led to the creation of a powerful form of collaboration known as the free software movement.

Stallman is quick to point out that "free software does not mean that the software is free, as in requiring no payment. When I speak of free software, I'm referring to freedom, not price. So think of free speech, not free beer."[6] Stallman applies four strict criteria to maintain free values in software:

0. The freedom to run the program, for any purpose (freedom 0).
1. The freedom to study how the program works, and adapt it to your needs (freedom 1). Access to the source code is a precondition for this.
2. The freedom to redistribute copies so you can help your neighbor (freedom 2).
3. The freedom to improve the program, and release your improvements to the public, so that the whole community benefits (freedom 3). Access to the source code is a precondition for this.[7]

Free software is distributed with the source code disclosed or open at the point of distribution. Non-free or closed software is distributed with no source code disclosed,[8] requiring anyone who wishes to discover that source code to engage in a

---

6 Richard M Stallman, "Free Software: Freedom and Cooperation", Speech at New York University, New York, 29 May 2001 <http://www.gnu.org/events/rms-nyu-2001-transcript.txt> (27 August 2001). On the power of free software models to enhance digital diversity consider: B Fitzgerald, "Intellectual Property Rights in Digital Architecture (including Software): The Question of Digital Diversity?" [2001] EIPR 121; B. Fitzgerald, "Software as Discourse: The Power of Intellectual Property in Digital Architecture" (2000) 18 Cardozo Journal of Arts and Entertainment Law Journal 337.

7 "The Free Software Definition", Updated 27 October 2001,

<http://www.fsf.org/philosophy/free-sw.html> (23 July 2002).

8 Although in some circumstances code vendors will allow inspection of their source code, and even building of the object code. See text at note 33 below.

process of reverse engineering by decompiling the machine code into source code. The fear that attaches to distributing the source code with software is that a recipient may use it to their advantage and profit without giving back to the community, free-riding on the community based developments. In order to remedy the most extreme examples of this Stallman ensured that the source code he distributed was covered by a lawfully binding obligation created through the GNU[9] General Public Licence (GPL).[10] The GPL provides that if you take free software code and create and distribute a new work based on the code, you are obliged to disclose your code to the people you are distributing to, which in essence means the whole community. In this way the GPL leverages upon the copyright in software code owned by the person licensing out the code to oblige the recipient to share improvements with the community for everyone's benefit.

This was Stallman's powerful insight: copyright in software code can be used not only to close access and exploit its benefits for monetary reward but can also be claimed at the source to structure open access down-stream.[11]

Copyleft and Non Copyleft Licences

There are two main types of free and open source software licences. The simpler licences, for example the revised[12] 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 copyleft licences, like the GNU General Public Licence (GPL),

---

9 "GNU" is a recursive acronym for "GNU's Not Unix". "A recursive acronym is an acronym which refers to itself in the expression for which it stands, similar to a recursive abbreviation. The earliest example is perhaps the credit card VISA, which was named in 1976 as a recursive acronym for VISA International Service Association. In computing, it soon became a hackish (and especially MIT) tradition to choose acronyms and abbreviations which referred humorously to themselves or to other abbreviations. Perhaps the earliest example in this context, from about 1977 or 1978, is TINT ("TINT Is Not Teco"), an editor for MagicSix.": <http://en.wikipedia.org/wiki/Recursive_acronym>

10 "The General Public License (GPL)", Version 2, June 1991, <http://www.opensource.org/licenses/gpl-license.html> at 19 August 2001.

11 For a detailed overview of and motivations for peer and user led production, of which free software is a prime example, see: Y Benkler, "Coase's Penguin, or, Linux and The Nature of the Firm" (2002) 112 Yale LJ 369; J Lerner & J Tirole, "Some Simple Economics of Open Source" (2002) 50 J. Indus. Econ. 197; E von Hippel, "Innovation by User Communities: Learning from Open Source Software" (2001) 42 Sloan Mgmt Rev 82. Benkler's greatest insight is that in certain circumstances peer production promises to provide the most efficient solution by bringing together the best intellects for the job. It can do this because it has the capacity to utilise a vast distributed network of knowledge which, in certain circumstances, is far superior than any other more formally or traditionally organised mode of knowledge production. Motivation to participate in sharing of knowledge through peer production he explains is on current evidence reasonably achievable due to "indirect appropriation" – money, design of the end product, pleasure or social profile gained through involvement in peer production.

12 The original BSD license had what came to be known as a 'obnoxious advertising clause', which required attribution to be displayed on all advertising materials. This caused a problem when there were many contributors to a project, because the attribution material quickly became large and unwieldy. Current versions of this license do not include the clause, but there are still many examples of software products released under the original license or modified versions of the original license.

attempt to create a contributory commons by requiring that any re-distribution of the software or its derivatives is released under the free licence.[13]

One other aspect that needs to be clarified at the outset is terminology. What is the difference between free software[14] and open source software?

Free Software v Open Source

The Open Source Initiative (OSI) is a non-profit organization. Its leading proponent, Eric Raymond, has conceptualised business models enabling commercial exploitation of open source programs.[15] Programs distributed with the Open Source Certified trademark (OSI Certified)[16] are published on an approved list of licenses[17] that conform to the open source definition.[18] The main elements of such licenses are:

- Free redistribution so that a party may not require a fee or royalty for the downstream distribution of the program;
- The program must include source code, and must allow distribution in source code as well as compiled form. If a program is not distributed with source code, there must be a well-publicized means of obtaining the source code for no more than a reasonable reproduction cost – preferably, downloading via the Internet without charge;
- Derived works and modifications must be allowed and be capable of distribution under the same terms as the original license,
- The license may preserve the integrity of the authors code by requiring that it be distributed as pristine base sources plus patches (modification chunks). In this way, "unofficial" changes can be made available but readily distinguished from the base source;
- The license must not discriminate against any person, group of persons, fields of endeavour, technology or software package;
- The rights attached to the program must not require entry to some other form of license or agreement such as a non-disclosure agreement,

---

13 See Lawrence Rosen, Open Source Licensing: Software Freedom and Intellectual Property Law (2004 Prentice Hall).

14 Note the term "freeware" – meaning free in price to download (see Trumpet Software Pty Ltd v OzEmail Pty Ltd (Australia) [1996] 560 FCA 1) – has nothing to do with the notion of making source code available for access and should not be used to describe the free software model

15 These include loss leader; widget frosting; give away recipe/open restaurant; accessorizing; free the future, sell the present; free the software, sell the brand; free the software, sell the content: Eric Raymond, The Cathedral and the Bazaar, <http://www.catb.org/~esr/writings/cathedral-bazaar>; Shane W Potter, "Opening Up to Open Source" (2000) 6 Rich. J.L &Tech 24; M Fink, The Business and Economics of Linux and Open Source (2002) Prentice Hall PTR

16 Open Source.Org, Revised 30 April 2001,

<http://www.opensource.org/docs/certification_mark.html> (24 November 2001).

17 Open Source.Org, <http://www.opensource.org/licenses/index.html>, (24 November 2001).

18 Open Source.Org, Version 1.9,

<http://www.opensource.org/docs/definition.html>, (20 July 2002).

- The rights attached to the program must not depend on the program's being part of a particular software distribution,
- The license must not place restrictions on other software that is distributed along with the licensed software. For example, the license must not insist that all other programs distributed on the same medium must be open-source software.[19]

The difference between open source and free software is at a philosophical level of abstraction. Because the definition of 'open source' is somewhat broader than the definition of 'free software', it is clear that all free software is open source, but not all open source software is free. In practice, however, most licences that satisfy the OSI definition will also be considered 'free'.[20]

In an effort to be all encompassing in discussion of this area of activity while respecting the nuances of the ideological differences it has become fashionable to use the term Free/Libre and Open Source Software (FLOSS), especially in Europe.

C. Free Software: A Requirement for Core Democratic Infrastructure

It is desirable that the operation of our governments be transparent: we need to have trust in the work of the State. In this paper we suggest that key government responsibilities that are moving to computer based management, such as elections and the administration of justice, should use free software as the primary software resource to ensure that transparency and trust. In this paper we focus on the need to employ free software when adopting code for use in government, taking the software used in the electoral process as a prime example.

i) The Approach of Governments to this Point

Today, nearly every government in the world wants to know more about free software and how the model works, and the private sector is not far behind. Some governments have already begun the task of migrating to the use of free software in

---

19 Ibid.

20 For an example of a OSI approved licence that is not considered 'free' by the Free Software Foundation, see the Reciprocal Public License <http://www.opensource.org/licenses/rpl.php> (at 4 January 2005). The FSF consider that this license is non-free because: "1. It puts limits on prices charged for an initial copy. 2. It requires notification of the original developer for publication of a modified version. 3. It requires publication of any modified version that an organization uses, even privately." (Free Software Foundation, 'Various Licenses and Comments about them', <http://www.gnu.org/licenses/license-list.html#NonFreeSoftwareLicense> at 4 January 2005). There are also licences which are considered free by the FSF and the OSI but not by some prominent free software groups, like Debian. See, for example, the 'Mozilla Public License' <http://www.mozilla.org/MPL/MPL-1.1.html> (at 4 January 2005), which Debian do not understand to be clearly free, and do not distribute in their main GNU/Linux distribution (Branden Robinson, Post to Debian-Legal Mailing List, 'Re: Bug#251983: libcwd: QPL license is non-free; package should not be in main' <http://lists.debian.org/debian-legal/2004/06/msg00131.html> at 4 January 2005; Barak Pearlmutter, Post to Debian-Legal Mailing List, 'OCAML QPL Issue', <http://lists.debian.org/debian-legal/2003/03/msg00459.html> at 4 January 2005; martin f krafft, Post to Debian-Legal Mailing List, 'Moving libcwd to Debian non-free' <http://lists.debian.org/debian-legal/2004/10/msg00009.html> at 4 January 2005).

the public sector. The free GNU/Linux operating system now rivals the dominance of Microsoft Windows in controlling how our computers and networks run, at least at an institutional level.[21] The Australian Government Information Management Office's (AGIMO) recognises that the use of open source software is "particularly widespread in areas such as network infrastructure, single-purpose computer servers, security, Internet and intranet applications and network communications" in both the private and public sectors.[22]

The press is full of stories of state movements to Free Software, for example the recent announcement that the Swiss government is switching to SUSE Linux for 3,000 of their servers. Operational efficiency and lower costs were cited as the primary reasons for the switch to SUSE Linux. 23 Novell is now making SUSE open source,24 in an attempt to develop a product that is open, Linux based, but easy to use. Mancusi-Ungaro, the Linux and OSS marketing director at Novell noted "OpenSUSE has a different goal [from RedHat's Fedora Foundation project]; it's all about end-user success. We have to develop something that's so usable that it can be deployed by someone who's not technical."25 This illustrates one of the problems that is attributed to Free Software, the technical requirements for using the code. Most people can install packaged proprietary software, but often there is an appearance of mystique for the installation or use of Free Software, even if this is a mis-attribution.

Europe

In Europe there has been a flurry of projects that are addressing the possibility of widespread adoption of Free Software. The FLOSS project (Free/Libre and Open Source Software)[26] ran from June 2001 for 16 months. It had European Commission funding to gather data on FLOSS use and development. The project was looking to find hard economic data on the effects of FLOSS contributions as a "non-monetary economic network", the distribution patterns of such software, measuring contribution and use, business models, particularly change management.[27] The project was remarkable as the first of its kind to collect such empirical data on a large scale on FLOSS. Following FLOSS's success at making an inroad into providing data on FLOSS, further EU projects have followed. FLOSS-POLS (Free/Libre/Open Source Software: Policy Support)[28] is a current project funded by the European Commission to analyse "government policy towards open source; gender issues in open source;

---

21 For example, Netcraft, a respected long-term Internet research and analysis organisation, in their most recent survey suggest that over 69% of all active websites use the free Apache webserver (Netcraft, June 2005 Web Server Survey <http://news.netcraft.com/archives/2005/06/01/june_2005_web_server_survey.html>);

22 AGIMO, A Guide to Open Source Software (2005), p 10.

23 Antony Savvas"Swiss government chooses Suse Linux" Computer Weekly, Friday 16 December 2005

24 http://www.opensuse.org/

25 China Martens "Novell to make SUSE Linux open-source" Computer World 3 August 2005

26 < http:// http://flossproject.org/ >. The project refers throughout to OS/F software as in "open source/free'.

27 ibid.

28 General description of project at < http://flosspols.org/ >

and the efficiency of open source as a system for collaborative problem-solving…. [and] focus on studying the impact of policy and providing policy recommendations."[29] By March 2005, FLOSS-POLS had surveyed 4,138 public authority IT administrators in 13 European member states (excluding Hungary). The key outcomes of this survey are that they found 79% of those surveyed used some FLOSS, and that there is a desire for increased use amongst them. Also notable is the different countries showed different profiles.

The CALIBRE (Coordination Actions for Libre Software),[30] project is also looking at FLOSS,[31] forming a research policy forum, and looking at coordination of FLOSS research, best practice and bringing these to industry.

The European Consortium for Open Source Software and Data in Public. Administration (COSPA) is a 4 million Euro project aimed at "introducing, analysing, and supporting the use of Open Data Standards (ODS) and Open Source (OS) software for personal productivity and document management in European Public Administrations (PA)"[32]

ii) Proprietary code

The purveyors of proprietary code have not been unconscious of the need for some access to code whether by business or government, but are often reluctant to make their code truly open. In some enterprise development environments, a partial solution can be provided by putting the code in escrow with a trusted third party, as a guarantee for access by the purchaser in cirtain circumstances. In government, there have been instances of a demand for access to the code, and the Free Software movement often cites this as a clear example of the advantage of non-proprietary, open, code. In response to this challenge, in 2003 Microsoft, announced its Government Security Programme (GSP) which makes its source code available for inspection by authorised personnel, in an authorised manner, even allowing for the building of the code. [33] However, given that Windows XP is estimated to have around 40 million source lines of code,[34] and the limited resources of government IT departments, it is unlikely that very much of the code would come under direct scrutiny under GSP.

In addition to the ability for creators of closed code to deliberately insert malevolent or 'anti-social' code (for example spy-ware), it is possible for the software to have more subtle effects. For example, something as simple as adopting a specific English

---

29 The survey report is at < http://flosspols.org/deliverables/FLOSSPOLS-D03%20local%20governments%20survey%20reportFINAL.pdf > and was completed on 14 July 2005.

30 Project details at <http://www.calibre.ie/ >. The project has funding of 1.5 million Euros under the European Union's Sixth Framework Programme.

31 Much of the CALIBRE documentation refers to FLOSS as simply 'libre'.

32 Details on the project are at http://www.cospa-project.org/ (accessed 28 July 2005)

33 The announcement "A Matter of National Security: Microsoft Government Security Program Provides National Governments with Access to Windows Source Code" at http://www.microsoft.com/presspass/features/2003/Jan03/01-14gspmundie.mspx

34 Gary McGraw "Managing Software Security Risks" IEEE Computer April 2002

spelling and grammar formats in a wordprocessor, or even in software dialogues, can clearly influence language and its use. It is also not surprising that firms that produce code will not embody socially beneficial aspects into the code if the firm thinks that it will not improve the bottom line of profitability.[35] Many commentators have argued that software, whether on the large scale as part of the internet,[36] or at the level of small granularity.

> Software is very much an agent of simulation in that it has tremendous capacity to reinvent reality through digitization. In understanding software and the regulation of it through law, we must be alert to its ability to distort and structure communication and to the power of those that create software to structure our identities through this process. [37]

Though software can be seen as a mirror on reality, its reflection also informs the structure of reality today. Only the obvious bias of software,[38] communications technologies, and complete systems, can be seen without detailed examination. There have been only few of these studies, such as on the design of educational software that is 'boy-biased', as is more appealing to males.[39] The "embedded normativity" of software is largely ignored.[40] It is clear that there is great potential for social engineering through the use of software, and not least in government systems. Free Software offers the potential for wide scale review that can expose such problems: at least is more likely than in proprietary code.

D. Case Study: The Electoral Process and E Voting Software

There are many well discussed benefits of free software,[41] although few address the issue of voting. In the interests of better efficiency in the voting system, reduced costs, faster results and greater accuracy, there has been a movement over many years to use more automation and computerisation in the election process.

Voting is a means of making a choice, and in democracies it is typically used as a means of determining the constitution of the government, and the modern democracy

---

35 See discussion of this aspect Jayp.Kesan and Rajiv C.Shah "Deconstructing Code" Yale Journal of Law &Technology 2003-2004 p277 at 378-382.

36 Such as Lessig L "The Law Of The Horse: What Cyberlaw Might Teach" Harvard Law Review [Vol. 113:501 and expanded elsewhere at length, and Johnson, D "Is the global information infrastructure a democratic technology?" Computers and Society 27 No.3 (1997) pp20-26

37 Brian F. Fitzgerald "Software As Discourse: The Power of Intellectual Property In Digital Architecture" Cardozo Arts and Entertainment [Vol. 18:337] (2000)

38 Such as the lack of useability of software by disabled people; So, C., Perry, M., Watt, S. "Towards an Accessible Web through Semantic Web Standards" proceedings of 2005 International Conference on Computers for People with Special Needs (CPSN'05).

39 Friedman B and Nissenbaum H "Bias in Computer Systems" in Huiman Values and the Design of Comptuer Technology (Cam. UP, 1997)

40 Brey P "Disclosive Computer Ethics" Computers and Society Dec 2000 p.10

41 See B Fitzgerald and Nic Suzor, "Legal Issues For the Use of Free and Open Source Software in Government" (2005) 29 *Melbourne University Law Review* 412; B Fitzgerald and G Bassett (eds.), *Legal Issues Relating to Free and Open Source Software* (2003) http://www.law.qut.edu.au/about/staff/lsstaff/fitzgerald.jsp

demands that elections are open and fair as well as being seen to be such. If citizens loose faith in their system of appointing their representatives they loose faith in their system of government. The Organization for Security and Co-operation in Europe has noted that in many countries the populace believes in some elections, whether or not the outcome is truly representative, the perception is that there were fraudulent practices in the operation of the democratic process.[42]

It has been suggested that the use of electronic voting systems may help in the security of the voting system, and provide better confidence in the voters in such systems. In the United States, for example, one of the purposes of the Help America Vote Act,[43] was to improve voting equipment, and although not mandated, many states have taken the opportunity to implement electronic systems.[44] Other nations have been early adopters of the electronic voting technologies, as Brazil that employed electronic devices to help in the voting process in 1990. The systems and framework for voting in Brazil has been developed to the extent that all-electronic elections were held in October, 2002. The voting machines use Microsoft Windows NT as an operating system, and touch screen functionality featuring pictures of the candidates.[45] Brazil's 2002 e-voting based elections were at that time the world's largest, with more than 115 million cast votes. However, in May 2004, India's elections involved more than half of 670 millions eligible voters who cast their vote electronically. 1.075 million electronic voting machines (EVM) were used in 543 Parliamentary and 697 Assembly Constituencies[46] Each Indian EVM can register up to 3840 votes, up to 16 candidates per balloting unit, and store the results for 10years. The use of the EVM was challenged in the courts, but this challenged rejected. The code for the EVMs is produced by two public sector companies in India, and is 'closed', the Frontline report quoting Venkataratnam of ECIL, one of the public sector companies, as saying: "The proof of the reliability of the programme, like any software, is in its repeated use without any flaws.".[47] However, the problems associated with the 'obfuscation' approach to security, where the source is kept hidden and closed is well documented. One problem is that an inadvertent leak of the code destroys this simplistic type of security permanently. This type of breach happened in the United States with some of the voting system code developed by Diebold through an insecure CVS server.[48] Relying on hiding the source code cannot

---

42 See reports of the Office for Democratic Institutions and Human Rights- Elections at http://www.osce.org/odihr-elections/14207.html, such as Haiti in 2000 or the Domincan Republic in 1994.

43 Help America Vote Act of 2002, 42 U.S.C.A. 15301-15545 [HAVA].

44 HAVA makes $325 million available for equipment. See also Daniel P. Tokaji "Early Returns on Election Reform: Discretion, Disenfranchisement, and the Help America Vote Act" 73 Geo. Wash. L. Rev. 1206

45 Spectrum OnLine 16 Aug 2004 http://www.spectrum.ieee.org/WEBONLY/resource/nov02/nbraz.html

46 Election Commission of India http://www.eci.gov.in/NewsLetter/ECI_NL_FEBJUN_2004.pdf.

47 R. Ramachandran "Tried, but tested?" Frontline Cover Story, Volume 21 - Issue 04, February 14 – 27.

48 Concurrent Versions System is a means of controlling source code and documentation for software development projects.

be regarded as an effective security measure. Good security in the software realm requires that the code withstand scrutiny. The Diebold incident was made public on *Scoop*[49] in 2003, allowing for serious analysis of the code by a group of computer scientists.[50] They pointed out several security loopholes and bugs in the code, which though Diebold offered rebuttals, did cast a shadow of mistrust. Another flaw with obfuscation by hiding the source code is that the final code, if accessible, may be reverse engineered to expose flaws. However, an independent 'Trusted Agent' report requested by the State of Maryland following the Hopkins Report supports the security viewpoint of Hopkins Report and others, but seems to suggest that the best security is not necessarily in the interests of commerce:

> "Good security principles dictate that the analysis of a system should presume that all components are publicly known. However, that does not imply that it is good practice to make those components known. In a commercial environment one must respect the rights to intellectual property that can provide a competitive edge. It can be argued, however, that subjecting source code to open scrutiny will not only motivate the programmers to write better code, but it will leverage the expertise of a much broader audience. It has the obvious downside of providing the malevolent user a blueprint of the system. Nevertheless, we are not aware of any in-depth source code analysis done on the Diebold software that matches the Hopkins team effort. "[51]

Such problems with code for electronic systems has led to many suggested technological solutions, whether by producing a paper receipt for the voter,[52] or providing other means for voters to 'check' that their votes were cast the way that they wished.[53] However, these do not address the underlying requirement that the code itself be trustworthy.[54] Electronic voting systems have much appeal. They can bring ease of use to voters, election candidates, and election officials. It can be argued

---

49 Bev Harris "U.S> Election Integrity Flaw Discovered At Diebold" Monday, 10 February 2003 < http://www.scoop.co.nz/stories/HL0302/S00052.htm >.

50 Kohno, T.; Stubblefield, A.; Rubin, A.D.; Wallach, D.S "Analysis of an electronic voting system" Security and Privacy, 2004. Proceedings. 2004 IEEE Symposium on9-12 May 2004 :27 – 40 [Hopkins Report]

51 RABA Innovative Solution Cell "Diebold AccuVote-TS Voting System" Trusted Agent Report for the Department of Legislative Services p8 (footnotes omitted).

52 Rebecca Mercuri "A Better Ballot Box?,", IEEE Spectrum, Volume 39, Number 10, October 2002.

53 Kaminski, H.; Kari, L.; Perry, M "Who counts your votes?" e-Technology, e-Commerce and e-Service, 2005. IEEE '05. Proceedings. The 2005 IEEE International Conference 2005 Page(s):598 - 603

54 It can also be argued that the hardware, too, should be 'open' as it also has the opportunity to 'interfere' with the voting system. The Pentium chip initial versions that were on the market contained a bug that caused the possibility for subtle calculation errors, Tom R. Halfhill "An error in a lookup table created the infamous bug in Intel's latest processor" Byte, March 1995 noted "Still, Pentium owners paid for a CPU that's supposed to perform floating-point math to IEEE standards, and that's not what they got. Instead, controversy has raged around additional issues: Intel's dismal customer relations and the wildly conflicting claims of how often the Pentium bug might bite a typical (nonscientific) user.

that electronic voting machines enhance the election process in many ways: they can handle multiple languages, they can easily adapt to the people with disabilities, and they provide faster results. However, there has also been much criticism of the use of electronic voting systems, especially those that operate on proprietary closed code systems, as the majority of those in use do. The opportunity for malicious code to subtly alter the outcome of an election is unprecedented with the use of computerised voting.

Within the Australian Capital Territory (ACT), a 1998 election had two candidate just three votes apart, and a recount that caught mistake in the original count…. Leading to demands for a computerized process to deal with both counting and voting. There were fifteen proposals received after a RFP, none of which satisfied the ACT Electoral Commission (ACTEC). However following a Request for Tender in December 2000, and seven tenders, the ACTEC chose EVACS, with work beginning in April 2001 for the system to be used at the elections in October 2001. Despite the short timeframe, the software was completed by Software Improvements Pty Ltd, tested and used successfully in the election.[55] The unique feature of this code is that it is written "using Linux open source software to ensure appropriate transparency",[56] and a 'minor error' was found through open sourcing the code and subsequently patched. The openness of the code is limited however, and has been criticized by the group that found the bug.[57]

In the UK there has considerable government investment in the construction of a secure electronic voting system.[58] Despite early indications that the UK government was interested in the development of electronic voting systems, even suggesting the ability to vote by 'texting' on a cell phone, there seems little interest from the Electoral Commission at this stage. The Department for Constitutional Affairs *Electoral Administration Policy Paper*[59] and responses from the UK Electoral Commission *Securing the vote* (published by the on 20 May 2005)[60] fail to address issues surrounding electronic voting, merely suggesting that should electronic voting be available and the 'technology allow', voters should be able to vote wherever they are. This is strange, as the report was in response to the Governments.

---

55 The history of EVACS development is documented by ACTEC at < http://www.elections.act.gov.au/EVACS.html >.

56 http://www.elections.act.gov.au/EVACS.html >.

57 The Logic and Computation Group at Australian National University with the National ICT Group comments: "However, only an incomplete representation of the eVACS code is available and there is no immediate way to build or test this code." < http://users.rsise.anu.edu.au/~rpg/EVoting/evote_conclusions.html >

58 Hansard 15 Jun 2005: Column 466W: "Ms Harman: The Government launched their electoral modernisation programme in 2002. This commenced a three-year programme of electoral pilots to test new methods of voting and other electoral innovations at local elections. The costs that the Government have incurred in connection with remote electronic voting pilots was £2,261,204 in 2002 and £16,976,000 in 2003."

59 The Department for Constitutional Affairs: Justice, Rights, Democracy "Electoral Administration a Policy Paper for Discussion" 26 May 2005, available at < http://www.electoralcommission.gov.uk/templates/search/document.cfm/13099 >

60 Available from the Electoral Commission, and independent body set up by the United Kingdom > Parliament < http://www.electoralcommission.gov.uk/templates/search/document.cfm/13101 >.

Electronic voting provides enormous potential for enhancing democratic ideals. However the integrity and accountability of such a system relies on the ability for people to be able to scrutinise and understand its process in a time efficient and effective manner. Voter Action in the United States reports recent attempt in New Mexico to "conduct meaningful inspections of their electronic voting machines" was rejected by election officials, [61] as was the request for examination of the files recorded from the 2004 presidential election, despite their being errors in polling in those elections on those machines. The voting officials claimed that opening the machines would invalidate the warranty by Election Systems and Software, and that "the machines store the results of public elections in a secret, proprietary format that ES&S claims as its private property". [62]

How do we know that the software employed to elect a President has not been coded by those with partisan interests. While such a claim may wreak of conspiracy paranoia, the fundamental role that software now plays in constructing and accessing knowledge and the wealth to be gained through technology means that we must aim to address these issues without delay. Our modest suggestion is that free software should be employed in core software infrastructure in a democracy to enhance participation in the development, and accountability, of such a discourse. [63]

E. IV. Software is a Governmental (and Governance) Process

Although bias and other undesirable effects of software **could** be part of **any** code they are more likely to be discovered and exposed in free software. Part of democratisation of society is the ability for each member of society to have a voice, and to have the opportunity to see how they are governed – participation in and access to knowledge. "Open Government" has been a catchphrase over the last decade, whether in Mongolia or the United Kingdom. [64] The OECD puts it thus:

> "Among the widely accepted principles of good governance are openness, transparency and accountability: fairness and equity in dealings with citizens, including mechanisms for consultation and participation; efficient and effective services; clear, transparent and applicable laws and regulations; consistency and coherence in policy formation; respect for the rule of law/ and high standards of ethical behaviour. There principles represent the basis upon

---

[61] *Patricia Rosas Lopategui v. Rebecca Vigil-Giron, et al* reported at http://www.voteraction.org/index.php/static/Lopategui_Lawsuit

[62] *idem.*

63 A. Massey ""But we have to protect our source!": How Electronic Voting Companies' Proprietary Code Ruins Elections" 27 Hastings Comm. & Ent. L.J. 233 at 253

[64] See http://www.cfoi.org.uk/opengov.html#og and http://www.open-government.mn/ .

> which to build open government – one that is more accessible, responsive and transparent in its operations."[65]

Such principles are equally applicable to the systems on which government depends for its operation.

Conclusion: The Politics of Knowledge and the Role of Technology

The democratic world clamours for "Free and Fair" elections, typically stimulated by perceived problems in paper based polls in developing nations.[66] It is our argument that technology like software has become integral not only to election systems, but also to the process of government and more broadly to governance. In any healthy democracy such a process should be understandable and open to review. What is more it should invite participation. We consider that free software provides a method for achieving more accountability and participation and should be utilised in core democratic infrastructure. As such the framework for accessing and constructing knowledge in the modern democracy informed by free software is open to be reviewed by any interested citizen. In any digital bill or rights or notion of digital constitutionalism[67] – the idea of regulating power relations in the digital environment – a core consideration needs to be the ability to access and understand the software infrastructure that gives meaning to core democratic processes. Free Software has the potential to be as, if not more, cost effective, secure, innovative, democratic and accountable than closed code software. This is not a wild scheme, as Free Software has already been adopted in many areas of government, business and society throughout the world.

---

[65] The Organisation for Economic Co-Operation and Development " Open Government: Fostering Dialogue with Civil Society" OECD 2003 France.

[66] Whilst this is being written, there are riots in Haiti, many claiming that the vote counting is not fair and honest.

[67] See further: B. Fitzgerald (ed) *Cyberlaw* Volume 1 (2005) Ashgate London; B. Fitzgerald, "Software as Discourse: The Power of Intellectual Property in Digital Architecture" (2000) 18 *Cardozo Arts and Entertainment Law Journal* 382–5; Paul Schiff Berman "Cyberspace and the State Action Debate: The Cultural Values of Applying Constitutional Norms to "Private' Regulation'" (2000) 71 *University of Colorado Law Review* 1263; Jack M. Balkin "Virtual Liberty: Freedom to Design and Freedom to Play in Virtual Worlds" (2004) 90 *Virginia Law Review* 2043; B Fitzgerald "Principles of Australian Constitutionalism" (1994) 1 (2) *Proceedings of the 49th ALTA Conference* 799; B Fitzgerald "Australian Constitutionalism" (20/6/97 Unpublished Manuscript on file with author); A Hutchinson, *Waiting for Coraf: A Critique of Law and Rights* (1995) University of Toronto Press, Toronto; *Associated Press v US* 326 US 1, 20. See further A Giddens, *The Constitution of Society* (1984) Polity Press, Cambridge; Alan Hunt *Foucault and law: towards a sociology of law as governance* (1994) Pluto Press, London; E Ehrlich *Fundamental Principles of Sociology of Law* (1936) trans. By WL Moll (NY: Arno Press edn 1975).