

International Free and Open Source Software Law Review

Volume 02 }

Issue 02 } December 2010

Editorials

- Editorial 103
Iain G. Mitchell, QC

Articles

- Licence Profile: Apache License, Version 2.0 107
Andrew Sinclair
- A Tale of Two Tragedies – A plea for open standards 115
Maurits Dolmans, with an introduction by Carlo Piana
- Project Harmony: Inbound transfer of rights in FOSS Projects 139
Amanda Brock
- Software Patents and Open Source Models in Europe: Does the FOSS community need to worry about current attitudes at the EPO? 151
Noam Shemtov
- Software Interactions and the GPL 165
Malcom Bain
- Affirmative action in procurement for open standards and FLOSS 181
Mathieu Paapst

Platform

- Software Package Data Exchange (SPDX™) Specification 191
Kate Stewart, Phil Odenice, Esteban Rockett
- Never a Better Time for Open Source 197
Matt Asay



About IFOSS L. Rev.

Editorial Committee

This publication is managed by a rotating Editorial Committee. The membership of the Committee for this issue is as follows:

Malcolm Bain

Malcolm is partner at id law partners, a Barcelona based law firm specialising in IT law, with a focus on FOSS projects. As well as his private practice, Malcolm participates pro bono in a number of FOSS related initiatives and teaches the legal aspects of FOSS at university.

Amanda Brock

Amanda Brock is General Counsel of Canonical, the commercial sponsor of the Ubuntu project. Having graduated with Honours from Glasgow University, Amanda went on to obtain a Masters in Comparative Jurisprudence from New York University Law School and a LL.M. in IT and IP law from Queen Mary and Westfield, University of London. She is author of E:Business; The Practical Guide to the Laws and has written extensively on commercial and IT law.

Richard Fontana

Richard Fontana is Open Source Licensing and Patent Counsel at Red Hat, Inc. Before joining Red Hat, Fontana was Counsel at the Software Freedom Law Center. Fontana is admitted to practice law in the State of New York.

Michel Jaccard

Michel is a partner in the technology and corporate practice groups of BCCC Avocats, an independent business law firm in Switzerland. He studied and worked in Lausanne, Geneva and New York, and taught and published extensively on legal issues relating to software licensing, information technology / business process outsourcing structures, data protection and online services. He regularly advises clients in Switzerland and abroad on intellectual property and technology as well as on corporate acquisitions, disposals and collaboration.

Andrew Katz

Andrew Katz studied Natural Sciences and Law at Cambridge University where he graduated with honours in 1989. In 1991 he was called to the Bar, and in 1993 requalified as a solicitor. He moonlighted as a programmer during his studies at Bar School, programming in Turbo Pascal. He has released software under the GPL. He is currently a partner at Moorcrofts LLP, a boutique law firm in England's Thames Valley and advises a wide range of businesses on free and open source related issues. He has lectured and published widely on the subject.

Iain G. Mitchell QC

Chairman, Scottish Society for Computers and Law; Chairman, Scottish Lawyers' European Group; Chairman, Faculty of Advocates IT Group; Lecturer, Honorary Board of Lecturers, Institut für Informations-, Telekommunikations- und Medienrecht, Westfälische Wilhelms-Universität, Münster; Freeman, Worshipful Company of Information Technologists.

Carlo Piana

Independent lawyer specialising in Information Technology and Telecommunication Law and Free Software Advocate. Started by using GNU/Linux and became intrigued by the legal and philosophical implications of it. Serves as Counsel to the Free Software Foundation Europe and advises projects and companies active in Free and Open Source Software.

Tomasz Rychlicki

Tomasz Rychlicki graduated from the University of Gdańsk, the Faculty of Law, Center of European Law. He also studied at Chicago-Kent College of Law in the LL.M. Program in International Intellectual Property Law. Tomasz currently works for PATPOL, Warsaw. He is a member of the Editorial Board at the Journal of Intellectual Property Law & Practice published by Oxford University Press, Oxford.

Brendan Scott

Brendan runs a legal practice based in Sydney, Australia. Brendan is a founding member and a director of Open Source Industry Australia Limited. He is a past president of the NSW Society for Computers and the Law and a past editor of its journal. He has over 15 years of experience in Technology and Telecommunications law and has a special interest in open source and the law related to it.

Ywein Van den Brande

Ywein Van den Brande (b. 1977) studied law, information science and economics at the Catholic University of Leuven and Groep T (Leuven, Belgium), the Université de Rouen (Rouen, France) and U.C.L.A. (Los Angeles, USA). Ywein is an active member of the Free Software Foundation Europe and is a member of workgroups on information architecture and usability. Ywein is the author of several legal contributions on law and technology, and regularly speaks at conferences. He is a guest lecturer at Groep T and blogs on the CIF blog of the journal www.tijd.be.

Martin von Haller Groenbaek

Martin von Haller Groenbaek is a founding partner at the leading Scandinavian IT law firm, Bender von Haller Dragsted. Among other things he specialises in open source, open content and open data licences. He is a co-founder of the Danish Open Source Vendors Association and of the Danish chapters of the Internet Society and Creative Commons. His blog is at <http://www.openlife.dk>. His online CV is at <http://www.linkedin.com/in/vonhaller>.

Martin von Willebrand

Partner, HH Partners Attorneys-at-law; Chairman, Validos Association, Chairman, Finnish IT law Association; Managing Director, Inventors Factory Association. Martin is a Finnish technology lawyer with particular specialisation in the areas of copyright, licensing and open source. He has authored and co-authored books on legal aspects of e-commerce, and has introduced collaborative models for delivering legal and other services.

Mark Webbink

Mark Webbink is a Visiting Professor of Law and Executive Director of the Center for Patent Innovations at New York Law School. Webbink is also a Senior Lecturing Fellow at Duke Law School and has served as an Adjunct Professor at NCCU Law School. From 2000 to 2007 Webbink served in various capacities with Red Hat, Inc., including General Counsel, Deputy General Counsel for Intellectual Property, Senior Vice President and Secretary. Webbink presently serves on the board of directors of the Software Freedom Law Center.

Editorial Coordinators

The editors wish to thank the Editorial Coordinators for their hard work and contribution to making the Review happen.

Graeme R. West (graeme@heliocentrik.net)

Graeme is a Masters student at the University of Edinburgh law school's SCRIPT IP and technology law centre, and is a former intern at the Free Software Foundation Europe. His academic background is in political science, in particular the global politics of IP. He has also worked in large-scale digital archival projects and web application development.

Shane Martin Coughlan (shane@opendawn.com)

Shane's academic background is in linguistics and politics, and his postgraduate research focused on cybernetic warfare. He founded FSFE's legal department in 2006, established the European Legal Network in 2007, and today focuses on improving efficiency in international technology markets.

Peer reviewers

The Editorial Committee wishes to thank the work of the many referees and peer reviewers whose professional expertise and dedication to high standards have made the publication of this issue possible.

Contact

All administrative, bibliographic and pre-publication enquiries should be directed to the Editorial Coordinators via email at:

admin@ifosslr.org

The Editorial Committee can be contacted via email at:

ed-com@ifosslr.org

Policies and bibliographic information

Copyright and licensing statement

IFOSS L. Rev. is committed to the improvement of understanding of legal issues in digital society. A licensing statement is therefore attached to each article, clearly outlining the particular terms which apply to the article. Most use Creative Commons licences with special exceptions for translations.

Graphic design

The Editorial Committee wishes to thank Tomasz Politański Design for its logo and associated graphic design work.

<http://tomaszpolitanski.com>

Publisher & sponsorship

IFOSS L. Rev. is published by its Editorial Committee, with financial and administrative assistance from NLnet Foundation and Mozilla Foundation. Please note that neither NLnet Foundation nor Mozilla Foundation accept correspondence on behalf of this publication. All correspondence should be directed to the Editorial Committee via email (see below).

Editorial policies

IFOSS L. Rev. accepts articles for publication from qualified personnel based on the criteria available to view on its web

site (<http://www.ifossl.org>). Submissions are welcome from all, and your business. Authors are strongly encouraged to read the style and content guidelines available on the web site. The review operates an anonymous peer review system for articles as appropriate, and expects all authors to meet the highest standards of scholarship and integrity.

Bibliographic information

The authors explicitly encourage libraries, archives and educational institutions to hold copies of IFOSS L. Rev. in their collections, in electronic and/or printed form. All users are advised that articles may occasionally be updated after publication. Linking back to original copies on the IFOSSL. Rev. web site, where authoritative versions are archived, is strongly recommended. Please contact the Editorial Coordinators for further information on best practices. It participates in the CrossRef system.

ISSN: 1877-6922

Publication schedule

IFOSS L. Rev. is published biannually. Submissions for publication are welcome at any time, but publication deadlines exist for each issue. For the latest information on papers sought and deadlines for submission, please consult the IFOSS L. Rev. website or contact the Editorial Coordinators at (admin@ifossl.org)



Yes, better, cheaper, faster. Open source will work, but what about the risks? **We can help.**

Your software developers are increasingly mixing open source components with proprietary code to lower project cost and reduce time-to-market.

The Black Duck® Suite allows developers to enjoy the productivity benefits of multi-source development while managing risk and complying with your company's open source policies.

Everything you want to know about Open Source from the Top Open Source Legal Experts.

Our informative webinar series has a wealth of content around open source from beginner to advanced topics.

You can view our library of recorded webinars and download the slides, and we continually address new topics. To be informed of upcoming webinars go to:



www.blackducksoftware.com

blackduck®

www.blackducksoftware.com/learnmore/

Editorial

Iain G. Mitchell Q.C.^a

on behalf of the IFOSS L. Rev. Editorial Committee'

*(a) Chairman, Scottish Society for Computers and Law;
Chairman, Scottish Lawyers' European Group; Chairman,
Faculty of Advocates IT Group; United Kingdom
representative on the IT Committee of the CCBE; Lecturer,
Honorary Board of Lecturers, Institut für Informations-,
Telekommunikations- und Medienrecht, Westfälische
Wilhelms- Universität, Münster; Liveryman, Worshipful
Company of Information Technologists.*

DOI: [10.5033/iffossr.v2i2.51](https://doi.org/10.5033/iffossr.v2i2.51)

Abstract

Iain G. Mitchell Q.C. looks back at the first two years of IFOSS L. Rev. sets the context for the exciting articles which appear in this issue and looks forward to the future.

Keywords

Law; information technology; Free and Open Source Software

Info

This item is part of the [Editorials](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement.

When the International Free and Open Source Software Law Review was launched in July, 2009, the Foreword stated:

"The Editorial Committee presents this first issue of the International Free and Open Source Software Law Review in the hope and expectation that it will provide a centre of excellence for the very best in analysis of issues facing users and advisors in the development, deployment and governance of Free and Open Source software, recognising the importance of digital rights issues to the daily professional and personal lives of many of the Review's readers and the role that open solutions might play in their resolution. The Review aims to present the perspectives of those most experienced and knowledgeable in the field and to ask how there might be attained sustainable solutions which foster the growth and development of the marketplace, whilst staying true to the underlying aim of enhancing digital freedoms for all."

Now as we enter 2011, we might ask whether and to what extent the Review has lived up to its promise.

There were those who questioned the sustainability of such a publication - after all, what is there to be said, not even about software licensing but, indeed Open Source Licensing? Is this not a very narrow field?

Well, the Review was launched into a changing world, and it is a pace of change which has not only been maintained, but has picked up speed in the intervening period, and with change, new questions present themselves and old ones come up again in very different contexts. This constant freshness is apparent from the range and scope of articles which have appeared.

The first edition carried the hopes of the Free and open Source licensing Community, not only in the terms of the editorial, but also in an article from Karen Faulds Copenhaver. At the time of the first Issue, the hot topic was the *Jacobson v Katzer* case, for here was a blessing from a U.S. Appeals Court of the open source licensing model. The review carried analysis of the case from both the U.S.¹ and English² perspectives and the hope seemed, as it still seems, justified. But the Review did not restrict itself to commentary and analysis as the first issue also launched the Risk Grid,³ a useful, practical tool for all of those involved in the field.

By the second issue, the perspective had broadened, both in scope, with articles considering the interaction of Open Source licensing with other forms of intellectual property, including Trademarks (the Tiki Dare and Harvey Anderson article),⁴ Patents and Open Standards (Sylvia Pfeiffer's *Techwatch* article)⁵ and also in time, looking back to the past (in Iain Mitchell's Article on *Hinton v Donaldson* in 1773 throwing old light on new controversies)⁶ and forward to the future (as in Susannah Sheppard's article looking forward to a future where open source solutions become *de facto* standards and analysing the issues that might cause with Competition Law).⁷

The Third Issue (Volume 2, number 1) saw new judicial support for Open Source Software, this time by the Italian Constitutional Court in relation to the freedom for a Regional Authority, consistently with procurement law, to favour open source software, and this provided material for a case report from Carlo Piana.⁸ The tradition for solid practical guidance established by the first issue was carried forward by Neil Brown's article on the GPL 2.0 and GPL 3.0 obligations to include licence text and provide source code,⁹ Martin Von Willebrand and Mikko-Pekka

1 Rosen, Lawrence (2009) 'Bad facts make good law: the Jacobsen case and Open Source', *IFOSS L. Rev.*, 1(1), pp 27 – 32,

2 Henley, Mark (2009) 'Jacobsen v Katzer and Kamind Associates – an English legal perspective', *IFOSS L. Rev.*, 1(1), pp 41 – 44

3 Coughlan, Shane; Katz, Andrew (2009) 'Introducing the Risk Grid', *IFOSS L. Rev.*, 1(1), pp 33 – 35

4 Dare, Tiki & Anderson, Harvey (2009) 'Passport Without A Visa: Open Source Software Licensing and Trademarks', *IFOSS L. Rev.*, 1(2), pp 99 – 110 DOI: [10.5033/iffossr.v1i2.11](https://doi.org/10.5033/iffossr.v1i2.11)

5 Pfeiffer, Dr Silvia (2009) 'Patents and their effect on Standards:Open video codecs for HTML5', *IFOSS L. Rev.*, 1(2), pp 131 – 138 DOI: [10.5033/iffossr.v1i2.21](https://doi.org/10.5033/iffossr.v1i2.21)

6 Mitchell QC, Iain G (2009) 'BACK TO THE FUTURE: Hinton v Donaldson, Wood and Meurose (Court of Session, Scotland, 28th July, 1773)', *IFOSS L. Rev.*, 1(2), 111 – 122 DOI: [10.5033/iffossr.v1i2.23](https://doi.org/10.5033/iffossr.v1i2.23)

7 Sheppard, Susannah (2009) 'Balancing free with IP: if open source solutions become de facto standards, could competition law start to bite?', *IFOSS L. Rev.*, 1(2), pp 73 – 82 DOI: [10.5033/iffossr.v1i2.16](https://doi.org/10.5033/iffossr.v1i2.16)

8 Piana, Carlo (2010) 'Italian Constitutional Court gives way to Free Software friendly laws', *IFOSS L. Rev.*, 2(1), pp 61 – 66 DOI: [10.5033/iffossr.v2i1.38](https://doi.org/10.5033/iffossr.v2i1.38)

9 Brown, Neil (2010) 'GNU GPL 2.0 and 3.0: obligations to include license text, and provide source code', *IFOSS L.*

Partanen's article on Package Review as part of the compliance process¹⁰ and Pieter Hintjens article on the Consensus-Oriented Specification system.¹¹ The forward-looking orientation was maintained by Luis Villa's Platform piece on Challenges and opportunities for open source legal communities.¹²

The trend which emerges from the first three issues is of a growing realisation of the way in which Open Source Software is not only developing but is also more and more becoming part of the mainstream as it continues to receive recognition from public authorities as well as courts and the wider business community. But if Open Source has well and truly broken out of the ghetto, that is going to throw into sharper relief the question of how it interacts with the wider world, technically, economically and legally. These questions, already flagged up in the first three issues, move on to centre stage in the present issue, which is, more than ever, about interoperability: Malcolm Bain looks at software interactions and the GPL;¹³ Noam Shemtov gives a review of the current approach of the EPO to Software patents,¹⁴ which serves to put into context the fraught issue of levelling the playing field for Open Source by compelling FRAND licensing of patents, a topic tackled by Maurits Dolmans;¹⁵ whilst the Procurement thread is carried forward by Mathieu Paapst in his article on affirmative action for Open Standards procurement.¹⁶ Lest all this seem a bit like looking at the trees rather than the wood, one could not do better than read Matt Assay's article, *Never a better time for Open Source*.¹⁷

And what of the future? The European Commission is now clearly convinced of the advantages of Open Source Software, and with the publication in December 2010 of the new European Interoperability Framework, the level of discussion, analysis and, it may be, litigation surrounding Interoperability looks set to gather pace. As Open Source increasingly becomes a serious contender in Public Procurement exercises, competitive restrictions, such as proprietary vendor lock-in will finally begin to be taken seriously, and, it may well be, will start to become the subject of litigation in both the procurement and competition fields. No doubt there will be much to be written, as time progresses, about how successful or not the Interoperability framework will have been.

In short, as the Review heads towards the start of its third annual volume, there will be no shortage of new things to write about and in respect of which to lead discussion, all as *the Review* has done

Rev., 2(1), pp 7 – 12 DOI: [10.5033/iffosslr.v2i1.31](https://doi.org/10.5033/iffosslr.v2i1.31)

10 von Willebrand, Martin and Partanen, Mikko-Pekka (2010) 'Package Review as a Part of Free and Open Source Software Compliance', *IFOSS L. Rev.*, 2(1), pp 39 – 60 DOI: [10.5033/iffosslr.v2i1.37](https://doi.org/10.5033/iffosslr.v2i1.37)

11 Hintjens, Pieter (2010) 'Consensus-Oriented Specification System', *IFOSS L. Rev.*, 2(1), pp 85 – 99 DOI: [10.5033/iffosslr.v2i1.32](https://doi.org/10.5033/iffosslr.v2i1.32)

12 Villa, Luis (2010) 'Lawyers and the Bazaar', *IFOSS L. Rev.*, 2(1), pp 77 – 84 DOI: [10.5033/iffosslr.v2i1.34](https://doi.org/10.5033/iffosslr.v2i1.34)

13 Bain, Malcolm (2010) 'Software Interactions and the GNU General Public License', *IFOSS L. Rev.*, 2(2), pp 165 – 180 DOI: [10.5033/iffosslr.v2i2.44](https://doi.org/10.5033/iffosslr.v2i2.44)

14 Shemtov, Noam (2010) 'Software Patents and Open Source Models in Europe: Does the FOSS community need to worry about current attitudes at the EPO?', *IFOSS L. Rev.*, 2(2), pp 151 – 164 DOI: [10.5033/iffosslr.v2i2.43](https://doi.org/10.5033/iffosslr.v2i2.43)

15 Dolmans, Marurits (2010) 'A Tale of Two Tragedies – A plea for open standards, and some comments on the RAND report', *IFOSS L. Rev.*, 2(2), pp 115 – 138 DOI: [10.5033/iffosslr.v2i2.46](https://doi.org/10.5033/iffosslr.v2i2.46)

16 Paapst, Mathieu (2010) 'Affirmative action in procurement for open standards and FLOSS', *IFOSS L. Rev.*, 2(2), pp 181 – 190 DOI: [10.5033/iffosslr.v2i2.41](https://doi.org/10.5033/iffosslr.v2i2.41)

17 Assay, Matt (2010) 'Never a Better Time for Open Source', *IFOSS L. Rev.*, 2(2), pp 187 – 190 DOI: [10.5033/iffosslr.v2i2.48](https://doi.org/10.5033/iffosslr.v2i2.48)

since its inception. We might modestly suggest that the promise made in the first editorial looks set to be fulfilled.

Iain G. Mitchell QC

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifosslr.org>.

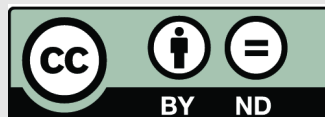
This article should be cited as follows:

Mitchell, Q.C., Iain (2010) 'Editorial', *IFOSS L. Rev.*, 2(2), pp 103 – 106
DOI: [10.5033/ifosslr.v2i2.51](https://doi.org/10.5033/ifosslr.v2i2.51)

Copyright © 2010 Iain G. Mitchell QC.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Licence Profile: Apache License, Version 2.0

Andrew Sinclair^a

(a) Legal Counsel, Canonical USA, Inc.

DOI: [10.5033/iffossr.v2i2.42](https://doi.org/10.5033/iffossr.v2i2.42)

Abstract

The Apache License, Version 2.0 is a widely used permissive open source software license which builds on the terms of other permissive open source licenses to produce a license with similar principles to those of licenses like the BSD License and MIT License, but with expanded terms such as a more developed copyright licence grant and a patent licence grant. This article is short overview of the license.

Keywords

Apache License; Law; information technology; Free and Open Source Software

Info

This item is part of the [Articles](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement. This article has been independently peer-reviewed.

As a permissive open source licence (an open source licence that features broad permissions and no “copyleft” provision), the Apache License, Version 2.0¹ (hereinafter referred to as the “Apache License”) has similar legal effects to those of licences like the BSD License, MIT License and historic permission notices. The Apache License is not as popular as these if measured by number of projects, but it is a significant open source licence due to the importance of the projects that use it, such as the Apache HTTP server, which is used by over 50% of all web servers.² The Apache License is also more developed than simpler licences like the BSD License. As a result, it relies less on common community interpretation to overcome potential legal ambiguities, while still retaining the core terms of a permissive open source licence.³

¹ <http://www.apache.org/licenses/LICENSE-2.0.html> [hereinafter *Apache License*]

² <http://news.netcraft.com/archives/2010/11/05/november-2010-web-server-survey.html>

³ For a discussion of potential legal ambiguities of the BSD License, see Sinclair, Andrew (2010) 'Licence Profile: BSD', *IFOSS L. Rev.*, 2(1), pp 1 – 6, DOI:[10.5033/iffossr.v2i1.28](https://doi.org/10.5033/iffossr.v2i1.28)

Licence history and use

The current version of the Apache License is the third iteration of licence. The history of the licence aids our understanding the current version: The original version was very similar to the original BSD License⁴, and the Apache Software Foundation updated the original to remove the required advertising clause following the same update of the BSD License.⁵ In 2004, the Apache Software Foundation approved version 2.0, which retains the same principles as the earlier versions, but expands and clarifies their terms.⁶

While it can be useful to gauge the importance of a licence by its popularity in terms of number of open source projects, such a gauge only measures one aspect of a licence's significance. In the case of the Apache License, popularity numbers have the potential to mislead. The Apache License comprises approximately 2% of the projects hosted on SourceForge⁷, but as the licence for all Apache Software Foundation projects⁸ and Android⁹, it is a licence that governs significant open source projects.

Parsing the licence

The Apache License consists of a series of clauses covering the licence terms followed by a short appendix setting out a standard format of a copyright and licence notice. This profile will discuss a few of the key clauses.

Copyright licence grant

As would be expected, the Apache License has a broad and very permissive copyright licence grant. It is surely written with the US Copyright Act in mind, as the grant language runs parallel to statutory rights enumerated in Section 106 of the US Copyright Act.¹⁰ The grant includes the rights to, “reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.”¹¹ Listing each of the US Copyright Act's exclusive rights in copyrighted works provides a far less ambiguous grant, at least under US law, than that of the BSD License or versions 1.0 and 1.1 of the Apache License. Those licences all have the same very simple grant, “Redistribution and use in source and binary forms, with or without modification, are permitted . . .”¹²

The copyright licence grants rights from each “Contributor”. This is a defined term roughly meaning anyone with a copyright ownership interest in the work or part of it. This structure,

4 Compare the original BSD License (<http://www.xfree86.org/3.3.6/COPYRIGHT2.html#6>) to the Apache License, Version 1.0 (<http://www.apache.org/licenses/LICENSE-1.0>)

5 <http://www.apache.org/licenses/> (Stating, “The 1.1 version of the Apache License was approved by the ASF in 2000. The primary change from the 1.0 license is in the 'advertising clause' (section 3 of the 1.0 license); derived products are no longer required to include attribution in their advertising materials, only in their documentation.”)

6 *Id.*

7 <http://www.oss-watch.ac.uk/resources/apache2.xml>

8 <http://www.apache.org/licenses/#distributions>

9 http://www.openhandsetalliance.com/android_faq.html

10 United States Copyright Act, 17 U.S.C. § 106 (2009)

11 *Apache License*, *supra* note 1

12 <http://www.opensource.org/licenses/bsd-license.php>

whereby the rights under the licence are granted from each copyright holder to each downstream licensee, is common in open source licensing, but it is not always explicit. One interesting feature of the Apache License grant is that it includes a right to sublicense. This may be intended to ensure that the Apache License is compatible with different licensing models. For example, if a licensee wanted to combine code licensed under the Apache License with code under a “copyleft” licence (a licence requires the resulting combination to be licensed under only that licence), that licensee may have no other choice but to sublicense the Apache License code under the “copyleft” licence. The sublicense right could also be useful in a proprietary licence context if a proprietary software vendor desires to sublicense Apache License code rather than simply passing the Apache License through to the end user or re-distributor. In many cases, however, there would not be a need to sublicense because the licence is granted directly from each “Contributor” to each licensee.

Patent licence grant

The Apache License expressly grants both a copyright licence and a patent licence to licensees. This is somewhat unusual among permissive open source licences, which do not usually mention patents. The delineation of the copyright and patent grants give clarity to the scope of licence which is absent in more simple examples like the BSD License and MIT License. Those licences are sometimes thought not to grant a patent licence, but their grant language does not mention copyright *or* patent. The licensee must look for clues in the context of the grant, such as references to the copyright statement or “copyright holders” elsewhere in the licence as well as community interpretation to assess whether a patent right is granted. The Apache License suffers no such ambiguity.

The patent licence grant of the Apache License, like the copyright licence grant, seems to be based on US statutory law. The licence includes the rights to make, use, sell, and import, which are the terms used in the US Patent Act.¹³ The licence, “applies only to those patent claims licensable by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted.”¹⁴ One of the problems with patent licences in open source software is that the software is likely to change downstream. A narrow patent license may not cover certain future modifications or combinations of the software, but a broad patent licence may license more than the licensor intended. The Apache License takes a relatively narrow approach; its grant is limited to the contribution and the resulting combinations of that contribution with the rest of the work at the date of submission. This grant seems to exclude patents that would be infringed by modifications of the contribution, at least to the extent such patents would not be infringed by the original contribution.¹⁵ The licensee receives the grant from all contributors, but if a midstream contributor altered upstream code or combined it with the work in a new way, the downstream licensee may not have received all the applicable patent rights from each contributor.

¹³ United States Patent Act, 35 U.S.C. § 271 (2010)

¹⁴ *Apache License*, *supra* note 1

¹⁵ The Apache Foundation License FAQ supports this reading, “Q1: If I own a patent and contribute to a Work, and, at the time my contribution is included in that Work, none of my patent's claims are subject to Apache's Grant of Patent License, is there a way any of those claims would later become subject to the Grant of Patent License solely due to subsequent contributions by other parties who are not licensees of that patent. A1: No.” (<http://www.apache.org/foundation/licence-FAQ.html>)

The last sentence of the patent licence section is not actually part of the licence grant; it is a patent termination clause. If a licensee under the Apache License brings a patent claim alleging that a work under the Apache License infringes that licensee's patent, the Apache License ceases with respect to that licensee. This is a simple and relatively strong clause, as it applies to any patent litigation claim with respect to code under the Apache License and does not offer any resolution period (the termination is effective immediately upon the filing of the patent litigation claim). Similar patent termination clauses can be found in other open source licences; one example is the Mozilla Public License version 1.1 (hereinafter "MPL")¹⁶. The MPL's patent termination clause terminates a potentially more narrow set of rights: those granted by the party against whom the claim is asserted, and it provides, in some cases, a sixty day resolution period giving the party bringing the claim an opportunity to resolve the dispute before the licence terminates.¹⁷ This clause should discourage patent claims where a would-be plaintiff is benefiting from the use of the Apache Licensed software.

Redistribution rights and conditions

The Apache License sets out a number of terms specific to redistribution.¹⁸ In addition to terms requiring preservation of legal and attribution notices, a distributor must provide a copy of the Apache License to recipients and add prominent notices with respect to modified files. These terms would be expected of a licence designed to replace the less specific BSD License, and provides practical guidance for those who may redistribute software under the Apache License.

The redistribution requirements section of the Apache License is the section that most frequently uses the defined term, "Derivative Work". This is likely due to the fact that most of the distribution requirements (e.g. the requirement to identify modified files) only become relevant with respect to derivatives. What is interesting though, is that the "Derivative Work" definition seems to be based on US statutory law (the US Copyright Act), but with modifications.

The Apache License definition omits the examples in the US Copyright Act and combines the two US Copyright Act sentences into a conjunctive requirement: the work must be both a "work based on" the original work *and* the modifications as a whole must represent an original work of authorship to be a "Derivative Work".¹⁹ This definition still defers to the law to answer important questions like whether the modifications are significant enough to constitute a derivative work, so it is not entirely clear what advantage the editing of the statutory definition provides. The second sentence in the definition is more helpful in guiding interpretation, as it sets out technical rules excluding certain software combinations from the derivative work analysis.

As a non-copyleft licence, the definition of "Derivative Work" is not of major significance. The Apache License confirms its non-copyleft status in the last clause of the redistribution section, clarifying that modifications and "Derivative Works" as a whole may be licensed under different licensing terms.

Contributions

¹⁶ <http://www.mozilla.org/MPL/MPL-1.1.html> [hereinafter *Mozilla Public License*]

¹⁷ *Id.*

¹⁸ *Apache License*, *supra* note 1

¹⁹ *Id.*

The Apache License has a notable clause dealing with contributions. The contributions clause, along with the contributions definition, provides clarity to open source projects that use the Apache License with respect to code submitted to the project without any separate agreement or licence notice. Such code is deemed to be licensed under the Apache License unless the author marks it with “not a contribution”. This is a practical feature of the Apache License, supporting a collaborative development model by removing licensing ambiguity with respect to informally submitted works. The clause is also flexible, expressly contemplating separate contribution agreements and permitting them to override the Apache License.

Compatibility

Apache and GPLv2

According to the Free Software Foundation, the Apache License 2.0 is not compatible with GPLv2 due to “certain patent termination and indemnification provisions.”²⁰ Both the Apache Foundation and the Free Software Foundation have updated their statements about compatibility over time, and a previous Free Software Foundation statement mentioned only “certain patent termination” provisions.²¹ A potential incompatibility in this respect results from the Apache License’s patent termination provision, which could be thought to restrict the licensee’s rights. The GPLv2 expressly prohibits “further restrictions” on the rights granted in the GPLv2.²² The Apache Foundation provided a detailed analysis of this argument, which left the compatibility question open.²³ However, the Free Software Foundation’s website still has a simple statement that it believes the Apache License 2.0 to be incompatible with GPLv2.²⁴ The current Apache Foundation statement does not directly challenge this, but instead defers to the judgement of Free Software Foundation, stating that “The Apache Software Foundation believes that you should always try to obey the constraints expressed by the copyright holder when redistributing their work.”²⁵ The analysis of whether the licences are compatible seems to hinge on an interpretation of GPLv2 rather than the Apache License, and authors of these licences seem to suggest that moving to GPLv3 is a better way to answer the compatibility question.

Apache and GPLv3

The issue of Apache compatibility was directly addressed in the drafting of GPLv3.²⁶ GPLv3 permits certain additional conditions to be added, one of which addresses the indemnity clause of the Apache License.²⁷ The indemnity clause applies when the a distributor offers additional

²⁰ <http://www.gnu.org/licenses/license-list.html> [hereinafter *FSF Statement*]

²¹ For older Apache Foundation statements about Apache License and GPL compatibility, see <http://web.archive.org/web/20060426193453/http://www.apache.org/licenses/GPL-compatibility.html> [hereinafter *Historic Apache Statement*]

For older Free Software Foundation statements about Apache License and GPL compatibility, see http://web.archive.org/web/*http://www.gnu.org/licenses/license-list.html

²² <http://www.gnu.org/licenses/gpl-2.0.html>

²³ *Historic Apache Statement*, *supra* note 24

²⁴ *FSF Statement*, *supra* note 23

²⁵ <http://www.apache.org/licenses/GPL-compatibility.html> [hereinafter *Apache Statement*]

²⁶ Free Software Foundation, *GPLv3 Final Discussion Draft Rationale*, p. 9 (available at <http://gplv3.fsf.org/rationale>)

²⁷ *Id.*

liability obligations to downstream recipients (e.g. a warranty or indemnity).²⁸ The Apache License permits these, but requires that the entity offering these indemnify upstream “Contributors” against claims asserted “by reason of” the additional obligations.²⁹ Interpreting the “by reason of” phrase caused the Free Software Foundation originally to view the clause as problematic, but a subsequent interpretation as “nothing broader or vaguer than ‘directly as a result of’” led to the Free Software Foundation’s agreement of GPLv3 compatibility (with the applicable additional condition mentioned above).³⁰ This view seems appropriate, as additional warranty or indemnity liability can be offered outside the context of the Apache License. For example, a licensor could offer a warranty as part of a negotiated commercial support agreement. The relevant clause of the Apache License seems designed only to protect the copyright holders from any additional liability imposed as a result of a commercial arrangement between a downstream licensee and its customer.

The Free Software Foundation made other changes to drafts of GPLv3 which addressed potential compatibility issues related to the trademark clause and patent termination clause of the Apache License.³¹ With regard to the trademark clause, the Apache License expressly reserves the trademark rights of the licensor, and GPLv3 expressly permits supplementing the GPLv3 with terms declining to grant certain trademark rights.³² These clauses demonstrate a conscious and successful effort to achieve compatibility between GPLv3 and the Apache License.

Conclusion

The Apache License can be thought of as a newer iteration of permissive non-copyleft licences like the BSD License, MIT License, and historical permission notice. In that role, the Apache License provides additional legal clarity to the permissive licensing model, and also adds new terms such as a patent termination provision. The Apache License is written in the context of US law, matching phrasing to the US Copyright Act and US Patent Act, which is helpful to increasing clarity with respect to licensing in the US.

Some of the Apache License terms which go beyond simpler licences in the non-copyleft family have lead to debates over its compatibility with other licences, but there is a consensus that GPLv3 and the Apache License 2.0 are compatible.

About the author

Andrew Sinclair is Legal Counsel at Canonical, the commercial sponsor of Ubuntu.

²⁸ *Apache License*, *supra* note 1

²⁹ *Id.*

³⁰ Free Software Foundation, *supra* note 29

³¹ *Id.*

³² *Id.*

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

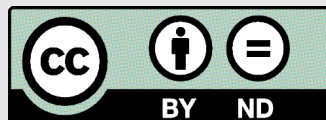
Sinclair, Andrew (2010) 'Licence Profile: Apache License, Version 2.0', *IFOSS L. Rev.*, 2(2), pp 107 – 114

DOI: [10.5033/ifosslr.v2i2.42](https://doi.org/10.5033/ifosslr.v2i2.42)

Copyright 2010 Andrew Sinclair.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



A Tale of Two Tragedies – A plea for open standards

Maurits Dolmans^a

*(a) Lawyer, Cleary Gottlieb Steen & Hamilton LLP, London
and Brussels offices, Rotterdam, New York and Brussels Bars*

With an introduction by Carlo Piana^b

*(b) Lawyer, Array, Milan, Italy, member of the Editorial
Committee of IFOSS l. rev.*

DOI: [10.5033/iffossr.v2i2.46](https://doi.org/10.5033/iffossr.v2i2.46)

Abstract

The IT sector is characterized by two market failures, the “tragedy of the commons” and the “tragedy of the anti-commons”, both of which must be resolved if IT innovation is to flourish and lock-in avoided. This involves a careful balancing of IPR protection and standard-setting, while avoiding hold-up and preserving opportunities for the significant innovation provided by the open source movement. The Author examines the shortcomings of the present system from a European Law perspective and expresses a plea for Open Standards in the interest of innovation and technological progress. As IT progresses, more and more products are compound items, incorporating technology co-owned by many different patent holders, co-manufactured by different producers, and interoperating with other complex products. A hold-up by a non-practicing entity or a rival using a single patent on a single component can kill an entire product. This article explores the criteria for “open standards”, and explains why royalty-free licensing of interoperability standards is appropriate in the software area (since RF standards can be implemented in both open source and proprietary software, thus allowing both models to compete on quality and functionality), while FRAND licensing is necessary for telecommunications. The notion of FRAND terms is further explored from a legal and economic perspective, explaining ways to determine fair pricing, and the need to ensure non-discriminatory terms in order to preserve competition in products implementing the standard. The article concludes with some comments on an interesting report by RAND Europe on “Trends in Connectivity Technologies and their Socio-Economic Impacts – Policy Options for the Ubiquitous Internet Society”.

Keywords

Law; Information Technology; telecommunications, Interoperability; Open Standards, Standard Setting, FRAND, RAND; Patents, hold-up.

Info

This item is part of the [Articles](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement. This article has been independently peer-reviewed.

Introduction By Carlo Piana

The article provides a very good explanation of why the interconnection between standardization and patents need a serious overhaul in order to address the concerns of competition failure in both the software and the general IT market, in the direction of what we call “Open Standards.”

On behalf of the Editorial Committee, I would like to profoundly thank the Review “Concurrences”,¹ where the article first appeared, for very kindly allowing republication of the same on our review. It is our policy to republish non-original works only in a few cases, when the relevance of the matter, the quality of the writing and of the analysis justify so; and when the first publication was on a review which has a separate readership from ours. Maurits Dolmans's article matched both requirements and brings a very valuable contribution to the discussion.

Maurits is a fine lawyer, a rigorous writer and an expert in the field. I have had the pleasure to work with him in a number of cases, first and foremost the seminal case Microsoft.² In this writing he analyzes the interaction between standard and patents, how the current IPR policies of the standards setting organization often fall short of delivering a truly open and independent standard. Conversely, the current practices, the intertwined and interdependency of modern IT and the nature of patents create more often than not inextricable “patent thickets” which serve no other purpose than extort unjustified royalties through questionable practices, under the common name of “patent hold-up.” I could not agree more.

The Author also analyzes remarkably well how – too frequently – RAND terms imply an actual discrimination against Free and Open Source Software implementations. Again, this is consistent with my experience, especially as a lawyer assisting clients in technology transfer agreements, mainly in the multimedia field.

Maurits adopts a position that clearly favors Open Standard without any ideological bias. He has a background that is very different from mine and those who are most likely to write on this Review. He finds for the case of Open Standards on a purely legal and technical point of view, out of a long and intensive experience on what RAND means in practical terms. Not ideology, indeed, but only simple and pertinent motives drive those who favor competition and technological advancement over hyper-exploitation of proprietary rights and an idolization of “Intellectual Property.”

¹ Concurrences N° 12010, n°30204, pp. 1338 <http://www.concurrences.com>

² Case T-201/04 *Microsoft v Commission* [2007] ECR II-3601



A Tale of Two Tragedies – A plea for open standards, and some comments on the RAND report

As early 1992, the European Commission published a Communication on Intellectual Property Rights (“IPRs”) and Standardization, requiring open access to European standards on irrevocable, fair, reasonable, and non-discriminatory (“FRAND”) terms.³ In 1999, an ETSI working group discussing a definition of FRAND for 3G mobile communications concluded that the Maximum Cumulative Royalty Rate “*should be set in single figures*”.⁴ The 2001 Guidelines for Horizontal Agreements require that “*an appreciable proportion of the industry is involved in the setting of the standard in a transparent manner,*”⁵ that “*the necessary information to apply the standard must be available to those wishing to enter the market,*” and that “*access to the standard must be possible for third parties on fair, reasonable and non-discriminatory terms.*”⁶ To summarize it all, Commissioner Kroes stated recently that “*Interoperability encourages competition on the merits between technologies from different companies, and helps prevent lock-in. Standards are the foundation of interoperability.*” She added that rates must be fair, and “*based on the inherent value of the interoperability information (rather than the information's value as a gatekeeper).*” She concluded: “*choosing open standards is a very smart business decision indeed.*”⁷

In this light, it is surprising that there is so much disagreement on the definition of open standards and on the conditions for licensing essential IPRs. These questions are quite contentious as

3 EC Commission, Communication on IPRs and Standardization, COM(1992) 445, section 6.2 (General Principles), available at <http://aei.pitt.edu/1222/http://aei.pitt.edu/1222/>.

4 Report by the UMTS IPR Working Group, “Third Generation Mobile Communications: The Way Forward for IPR”, January 1999, available at http://www.3gpp.org/ftp/PCG/PCG_01/Docs/PCG1_11.pdfhttp://www.3gpp.org/ftp/PCG/PCG_01/Docs/PCG1_11.pdf. See also Ericsson’s Comments on the European Commission’s White Paper on ICT Standardisation, available at http://ec.europa.eu/enterprise/sectors/ict/files/consultation_standardisation_2009/128_ericsson_en.pdfhttp://ec.europa.eu/enterprise/sectors/ict/files/consultation_standardisation_2009/128_ericsson_en.pdf.

5 European Commission Notice - Guidelines on the applicability of Article 81 [now Article 101 TFUE] to horizontal co-operation agreements, OJ C3, January 6, 2001, at 25, para. 169 (2001) (“Guidelines on Horizontal Agreements”).

6 Guidelines on Horizontal Agreements, above, para. 169.

7 N. Kroes, SPEECH/08/317, “Being open about standards”, Speech to Open Forum Europe, Brussels, June 10, 2008, (“OFE Speech”). See also Commissioner Kroes, SPEECH/09/475 “Setting the Standards High”, October 15, 2009. Interoperability is defined in the CFI Judgment in *Microsoft* (see below), para. 225 and following: “*interoperability between two software products means the capacity for them to exchange information and to use that information mutually in order to allow each of those software products to function in all the ways envisaged.*” This case concerned client-to-server and server-to-server communication, i.e., communication between two separate computer systems from different vendors in a network. See also para. 237: “*the attainment of that objective assumes that non-Microsoft work group server operating systems are capable of receiving a specific message from a Windows client PC or work group server operating system and giving the required response to that message on the same conditions as a Windows work group server operating system and also of enabling Windows client PC or work group server operating systems to react to that response just as though it came from a Windows work group server operating system.*” “Interoperability” between two computer systems should be distinguished from “compatibility” (whether a software or hardware component of a computer system can be substituted by another component without modification) and “portability” (whether a software or hardware component of a computer system or piece of software can be modified or adjusted to become part of another computer system). The Microsoft judgment did not cover those notions.

indicated by a series of recent debates and cases (such *EIF*,⁸ *Microsoft*,⁹ *Rambus*,¹⁰ *Qualcomm*,¹¹ and *IPCom*¹²). The debate is on occasion conducted with fundamentalist fervor, pitching proponents of unlimited IPRs on one side of the spectrum against the open source community on the other. It is useful, therefore, to analyze the debate from a coolly rational policy perspective, and ask whether the rules could not be adjusted to accommodate different situations in different industry sectors.

At the root of the problem is a conflict between measures to resolve two different market failures: the “tragedy of the commons” and the “tragedy of the anti-commons”.

We all know the “tragedy of the commons”, the overuse of public goods controlled by no one. In the 18th Century, it was found that common land in Britain was overexploited, because each user had an individual interest in letting the maximum number of cattle freely graze on it, with the result that the fields were exhausted, and everyone suffered.¹³ Even today, we poison ourselves or even risk changing our climate, because we produce goods the price of which does not include the cost imposed on society caused by the pollution of “free” air, soil and water (a “price externality”). Private restraints or public regulation may be needed to solve this market failure. Similar thinking led to the adoption of intellectual property laws. Innovators invest and sink funds in the creation of ideas and their expression. If all ideas were free and we all enjoyed full freedom to copy expression, imitators could enter the market without limitation, free riding on the innovators’ investments. To allow creators to raise price above marginal costs for a while and thus recover sunk R&D costs and be compensated for risk, IPRs exclude competition from imitators for some time, subject to conditions. Thus, patent law “*secures to the inventor, for a limited time, the exclusive use of his invention; and thereby adds the fuel of interest to the fire of genius.*”¹⁴

8 Draft European Interoperability Framework, available at

<http://ec.europa.eu/idabc/en/document/3473> <http://ec.europa.eu/idabc/en/document/3473>.

9 Case T-201/04 *Microsoft v Commission* [2007] ECR II-3601, and *Microsoft Interoperability Undertaking*, December 16, 2009, available at

<http://www.microsoft.com/presspass/presskits/eumsft/docs/MicrosoftInteroperabilityUndertaking16Dec2009.doc> <http://www.microsoft.com/presspass/presskits/eumsft/docs/MicrosoftInteroperabilityUndertaking16Dec2009.doc>. For ongoing debate on Microsoft’s activities in connection OOXML standard setting, see entries in Rob Weir’s blog, *An Antic Disposition*, available at <http://www.robweir.com/blog/>, especially “The Final OOXML Update”, parts I, II and III.

10 European Commission, Notice art. 27(4) Reg. 1/2003, *Rambus*, case COMP/38636, OJ C133, June 12, 2009, p.16; *Rambus* Article 9 Reg. 1/2003, commitments, available at

<http://ec.europa.eu/competition/antitrust/cases/decisions/38636/commitments.pdf>; See also press release IP/09/1897, December 9, 2009, and Commissioner Kroes, SPEECH/09/575, “Lessons learned for standardization”.

11 *Broadcom Corp. v. Qualcomm Inc.*, 501 F.3d 297, 310 (3d Cir. 2007) (“Deception in a consensus-driven private standard-setting environment harms the competitive process by obscuring the costs of including proprietary technology in a standard and increasing the likelihood that patent rights will confer monopoly power on the patent holder[...] Deceptive FRAND commitments, no less than deceptive nondisclosure of IPRs, may result in such harm”); See also Japan FTC *Cease and Desist Order Against Qualcomm*, September 20, 2009 (on appeal) available at <http://www.jftc.go.jp/e-page/pressreleases/2009/September/090930.pdf> <http://www.jftc.go.jp/e-page/pressreleases/2009/September/090930.pdf>, Korean FTC Press Release “KFTC took corrective measures against Qualcomm for abusing its monopoly market status in modem chip market; Imposition of fine and issuance of corrective order for discriminative royalty rates, conditional rebates, etc”, July 23, 2009; “EC closes formal proceedings against Qualcomm”, MEMO/09/516 of November 24, 2009.

12 European Commission, MEMO/09/549, December 12, 2009 on IPCom’s public statement confirming its FRAND Declaration, and IPCom statement http://www.ipcom-munich.com/IPCom_Frand_Declaration.pdf http://www.ipcom-munich.com/IPCom_Frand_Declaration.pdf.

13 G. Hardin, “The Tragedy of the Commons”, *Science*, Vol. 162, No. 3859 (December 13, 1968), p. 1243-1248

14 A. Lincoln, “Lecture on Discoveries and Inventions”, in *Collected Works of Abraham Lincoln* (R. Basler, ed., 1953)

Less well known is the opposite phenomenon, the “tragedy of the anti-commons.” This is the under-use of private goods that are controlled by more than one rightholder. Michael Heller in his fascinating *Gridlock Economy* mentions a series of arresting examples.¹⁵ Just a few: the banks of the Rhine are dotted with a sequence of picturesque robber baron castles, each of whom raised tolls on Rhine river traffic, with the result that no one used the river and no one received toll revenues. Google Book Search is being blocked by a multiplicity of rightholders in Europe, with the foreseeable outcome that orphan works remain dead, neither Google nor the European rightholders or libraries will make any money, and readers are deprived of access. And with every 3G cell phone being covered by thousands of patent families, each essential patent owner can block every other one, which the risk that no technology owner can use its technology without striking a compromise with the others.¹⁶

This is where standards and open source come in. As IT progresses, more and more products are compound items, made of many components incorporating technology co-owned by many different patent holders, manufactured by a series of different producers, and interoperating with other complex products. The computer and the mobile phone are perfect examples. They are multipart combinations of software, processors, modems, and electrical components, and are in turn part of networks that include other complex products (servers, base stations, routers, switches, servers and related network products). If each component manufacturer chose its preferred technology, these products could not interoperate in a network and no one could compete (or in exceptional cases, a manufacturer could gradually monopolize all key products in a closed network, chilling innovation¹⁷).

By compromising and selecting a *common standard*, producers are able to break the logjam. If they are truly open, standards allow “best of breed” components from different manufacturers to be combined, with maximum efficiency. Creating a standard, however, raises the risk of a tragedy of anti-commons at a higher level: patent “hold-up”. Once an industry has agreed to a standard, and especially after producers have implemented it, industry becomes “locked in”.¹⁸ This is an

(1858). An alternative to using IPRs would be for government, academia, or charitable institutions to fund R&D, or to look for alternative revenue opportunities such as services-funded or advertising-funded R&D.

15 M. Heller, *The Gridlock Economy – How Too Much ownership Wrecks Markets, Stops Innovation, and Costs Lives*, Basic Books, 2008.

16 This, too, is not a new concept. In economics, this is called a problem of “Cournot complements”, named after the 19th century French economist who discovered that monopolist producers of complementary products may both block each other to extract monopoly rents, thus reducing output below the level that a single monopolist would have produced. See M. Lemley and C. Shapiro, “Patent Holdup and Royalty Stacking,” (2007) *Texas Law Review*, Vol. 85:1991-2049, at <http://faculty.haas.berkeley.edu/SHAPIRO/stacking.pdf> <http://faculty.haas.berkeley.edu/SHAPIRO/stacking.pdf>, and C. Shapiro, “Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard-Setting”, May 2000, available at <http://ideas.repec.org/p/cla/levarc/122247000000000539.html> <http://ideas.repec.org/p/cla/levarc/122247000000000539.html>, J. M. Buchanan and Y. J. Yoon, ‘Symmetric Tragedies: Commons and Anticommons’, *Journal of Law and Economics* 2000, 43(1), 1–14.

17 Microsoft’s Interoperability Undertakings of December 16, 2009 are intended to resolve this lock-in issue. See above, footnote 9.

18 *Broadcom Corp. v. Qualcomm Inc.*, 501 F.3d 297, 310 (3d Cir. 2007), III.A.2.b.: “Although a patent confers a lawful monopoly over the claimed invention [...] its value is limited when alternative technologies exist [...] That value becomes significantly enhanced, however, after the patent is incorporated in a standard [...] Firms may become locked in to a standard requiring the use of a competitor’s patented technology. The patent holder’s IPRs, if unconstrained, may permit it to demand supracompetitive royalties.” See also C Madero Villarejo and N Banasevic, “Standards and Market Power”, *Global Competition Policy*, May 2008, 3.

opportunity for unscrupulous IPR owners whose intellectual property is essential for the standard. In the example of 3G cell phones mentioned above (with about 6,000 patent families for the air interface alone), even if all patent owners agree to license, each may have an incentive to threaten manufacturers with an injunction, a threat to kill their business, to extract extortionist fees – the commercial counterpart of Dick Turpin’s “Your money or your life”. A hold-up using a single patent on a single component can kill an entire product. Because cooperative innovation in today’s IT and telecom products fragments technology ownership (the 6,000 patent families mentioned above are owned by dozens of different firms), this risk is multiplied many times. If one patent holder makes his get-away after holding up another, others are encouraged to do the same.

When first confronted with this problem, industry players responded by developing a portfolio of countervailing patents. This is like an arsenal of nuclear missiles, with everybody pointing a missile at everyone else. If one player asserts patent rights against another, it is immediately counter-sued by the accused party. Firms even enter into mutual defense pacts.¹⁹ This cold war situation of Mutually Assured Destruction is hardly ideal. Too much effort goes into mining patents on too many trivial ideas. More important, it is no defense against patent trolls, non-vertically integrated firms that mine patents or buy them up from bankrupt estates to join a feeding frenzy, and who have nothing to lose from a counter-suit, since they do not engage in production.²⁰ The *IPCom* and *Rambus* cases come to mind.²¹ In a situation like this, strength becomes a weakness, since the largest producers are most vulnerable to the smallest opportunistic patent holder. This asymmetrical warfare even creates opportunity for strategic behaviour, where manufacturers are tempted to finance third-party IPR litigation against competitors,²² or to spin off patent portfolios to kill rivals or hold them up with a view to raising rivals’ costs.²³ If you can’t beat them, join them. Impenetrable patent thickets exacerbate the problem, because it becomes too

19 Examples include the open Innovation Network (OIN, <http://www.openinventionnetwork.com/>), the Allied Security Trust (<http://www.alliedsecuritytrust.com/>), Intellectual Ventures (<http://www.intellectualventures.com/>) and RPX Corporation (<http://www.rpxcorp.com/>). While many of these may be beneficial, the model is not without risk to industry and consumers, since some of the profit-oriented entities might eventually be tempted to evolve into a Ponzi scheme, start holding up industry members that have not joined them, or resell the patents to third parties that do so.

20 Trolls are sometimes called “non-practicing entities”. New technology business models such as patent trading and “patent mining” are not necessarily bad. The existence of a market for patents may foster innovation, and allow firms or groups like the Open Innovation network to acquire patents for defensive purposes. At the same time, patent traps and royalty traps (“hold-up”) may discourage investment, where remuneration is taken away from the person who incurred R&D costs and bore the risk of product development – in a situation where bringing products to market may well be more costly, more risky and more beneficial to consumers. The key is to intervene to prevent inefficient hold-ups, including “*opportunistic behavior on the part of patent owners that threatens to impose (1) static deadweight losses that are not justified by likely increases in dynamic efficiency, or (2) dynamic efficiency losses due to reduction in the incentive to participate in standard setting organizations or to engage in follow-up innovation.*” See T. Cotter, “Patent Holdup, Patent Remedies, and Antitrust Responses”, *Journal of Corporations Law*, July 1, 2009, available at <http://www.allbusiness.com/legal/civil-procedure-injunctions/12938773-1.html>.

21 See above, footnotes 11 and 13.

22 A well-known example is Microsoft’s financing of SCO to assert copyrights against Linux. See discussion at Groklaw, <http://www.groklaw.net/staticpages/index.php?page=20061212211835541>.

23 An interesting example is Microsoft’s attempt to quietly auction off 22 Linux-focused patents to non-vertically integrated patent companies, explaining how these could be used to against Linux, which was unexpectedly thwarted by Allied Security Trust buying them and reselling them the Open Innovation Network. See Groklaw, “Microsoft and A Patent Checkmate of My Dreams”, available at <http://www.groklaw.net/articlebasic.php?story=20090908164954318>.

difficult to challenge each and every patent used in the hold-up. The result can stifle innovation, standardization, and implementation of standards, for fear of future hold-up.

To address this problem, courts can take off the hard edges of intellectual property, as the US Supreme Court did in *Quanta v. LGE* (expanding the reach of exhaustion rules) and *eBay v. MercExchange* (limiting availability of injunctive relief subject to principles of equity, in case of patent suits by non-practicing entities), and as the German Federal Supreme Court did in *Orange Book* (limiting availability of injunctive relief under competition law where the dispute is merely about the amount of a FRAND royalty).²⁴ Legislatures can adjust patent laws, as the EC is proposing to do by proposing “license of right” arrangements.²⁵ Private parties cooperate, agreeing to avoid patents altogether as the open source community does,²⁶ requiring royalty-free licensing as W3C has chosen to do,²⁷ by encouraging patent pools as DVB does,²⁸ by agreeing to mutual price restraints as ETSI tried to do with its duty to disclose IPRs and license them on FRAND terms before finalizing the standard,²⁹ or by the use of competition laws or regulation.

24 *Quanta Computer Inc. v. LG Electronics, Inc.*, 453 F. 3d 1364; *eBay Inc v. MercExchange, L.L.C.*, 547 U.S. 388 (2006); *Bundesgerichtshof KZR 39/06*, decision of May 6, 2009.

25 See Revised proposal for a Council Regulation on the Community Patent, April 7, 2009, Article 20, available at <http://register.consilium.europa.eu/pdf/en/09/st08/st08588.en09.pdf><http://register.consilium.europa.eu/pdf/en/09/st08/st08588.en09.pdf>. The “license of right” is voluntary, and to have practical impact, additional incentives are probably required to encourage patent holders to agree with the license of right regime, for instance by facilitating the challenge of injunction patents, and by allowing “license of right” patents a greater presumption of validity. Moreover, defensive suspension of the “license of right patent” should be introduced, so as to allow defensive use. For some interesting articles see e.g. Boldrin and Levine “The Case Against Intellectual Property”, (2002) *American Economic Review* 92(2): 209-212 and M.A. Lemley, “A Cautious Defense of Intellectual Oligopoly With Fringe Competition”, (2009) *Review of Law & Economics*, Vol. 5 : Iss. 3, Article 3.

26 “Open source software is software that is distributed under an open source license. The open source license gives anyone who is interested the right to access the program's source code and to copy, modify, and redistribute the program on a royalty free basis. There are many different open source licenses, but these characteristics are common amongst all of them. The most popular open source software programs also use an open source development methodology. An open source methodology provides any interested programmer with access into the program development process, and a democratic, open means for development and enhancement of the program. Software can be open source even if the developers do not adopt an open source development methodology. Open source software is complementary to, and is often included in, commercial software. [...] Open source software can be an important source of innovation because it brings together people from different backgrounds and perspectives to work on and solve common business and IT problems. It is also an excellent approach for driving emerging standards and, in many cases, an open source software project can become the common implementation of a standard that is used by a large number of IT vendors and customers.” See IBM paper “Open Standards, Open Source, Interoperability and Government Policy”, May 11, 2009, <http://www.marketwire.com/press-release/Ibm-NYSE-IBM-902622.html><http://www.marketwire.com/press-release/Ibm-NYSE-IBM-902622.html>.

27 See W3C Patent Policy, available at <http://www.w3.org/Consortium/Patent-Policy-20040205/>.

28 See DVB Project promotes Pooling of DVB Patents, May 29, 1997, available at http://www.dvb.org/documents/press-releases/pr037_promotes%20Patent%20Pooling.pdfhttp://www.dvb.org/documents/press-releases/pr037_promotes%20Patent%20Pooling.pdf. See also C. Shapiro, “Navigating the Patent Thicket: Cross Licenses, Patent Pools, and Standard-Setting”, May 2000, available at <http://ideas.repec.org/p/cla/levarc/12224700000000539.html><http://ideas.repec.org/p/cla/levarc/12224700000000539.html>.

29 See ETSI IPR Policy and Guide on IPRs, available at <http://www.etsi.org/WebSite/AboutETSI/IPRsInETSI/IPRsInETSI.aspx><http://www.etsi.org/WebSite/AboutETSI/IPRsInETSI/IPRsInETSI.aspx>. See also “The Way Forward for IPR”, above, footnote 4. M. Lemley, “Ten Things to Do About Patent Holdup of Standards (and One Not to)”, (2007) 48 *B.C. L. Rev.* 149, 151-55 (2007). For an overview of IPR Policies, see also A. Updegrove, *The Essential Guide To Standard*, Chapter 4, Intellectual Property and Standard Setting, available at <http://www.consortiuminfo.org/essentialguide/intellectual.php><http://www.consortiuminfo.org/essentialguide/intellectual.php>.

These solutions tend to be effective only if they result in truly “open standards”. To qualify as “open”, the process for standards adoption, quality, and access to the standard must meet a number of conditions, all of which are needed to eliminate the tragedy of the anti-commons.³⁰ The following list includes several criteria:³¹

a. Open access to the decision-making process. No interested party should be excluded, unless on the basis of published, objective, relevant, proportionate, and verifiable criteria for admission.³²

*b. Open (transparent and undistorted) procedures.*³³ Governance rules for standards bodies should ensure that technology decisions, voting, and dispute resolution are representative, objective, and protected from undue influence. Vote stuffing or procedural irregularities such as alleged with respect to OOXML, for instance, disqualify a standard as open.³⁴

c. Open (published, pro-competitive) goals. Standards unnecessary for or not reasonably related to clearly defined, legitimate objectives such as interoperability, are naked restraints of inter-technology competition, and should not be allowed.³⁵

30 This is not to suggest that closed standards are prohibited. Joint R&D not intended to create an industry standard may meet the conditions of the Joint R&D Block Exemption Regulation No. 2659/2000, OJ L 304/7 (2000) or the *Guidelines for Horizontal Agreements*, above. Similarly, proprietary standards fairly achieved may be beneficial in fostering radical “break-out” innovation like Sun’s Java and the very notion of cloud computing to “escape” the local desktop or local network. Thus, open standards should not be mandatory, so long as the owner of the proprietary standard – if and when it prevails and develops network effects – is curbed from using its dominance to stifle “break-out” innovation.. Also, closed standards should not be given any preference in procurement, and may not obtain all the benefits of European standards.

31 See also European Commission, White Paper, Modernising ICT Standardisation in the EU - The Way Forward, COM(2009) 324 final, July 3, 2009, p. 4-6; IBM paper “Open Standards, Open Source, Interoperability and Government Policy”, May 11, 2009, and “IBM Announces new IT Standards Policy”, September 23, 2008, <http://www.marketwire.com/press-release/Ibm-NYSE-IBM-902622.html> <http://www.marketwire.com/press-release/Ibm-NYSE-IBM-902622.html> and <http://www.sutor.com/newsite/blog-open/?p=2615> <http://www.sutor.com/newsite/blog-open/?p=2615> ; Microsoft Interoperability Principles, February 21, 2008, <http://www.microsoft.com/interop/principles/default.mspx> <http://www.microsoft.com/interop/principles/default.mspx> ; and “The Meaning of Open”, in The Official Google Blog, December 21, 2009, <http://googleblog.blogspot.com/2009/12/meaning-of-open.html> <http://googleblog.blogspot.com/2009/12/meaning-of-open.html> . For an in-depth discussion of these criteria from a competition law perspective, see also M. Dolmans, “Standards for Standards” Vol 26, *Fordham Int’l L. J.* number 1, November 2002, p. 163-208 and “Standard Setting – The Interplay with IP and Competition Laws – How to avoid false FRANDs”, 2008 Fordham IPR Conference, in Hugh C. Hansen (ed.), *Intellectual Property Law and Policy*, Volume 12 (forthcoming).

32 See Guidelines on Horizontal Agreements, above, footnote 6. See also European Commission Decision 87/69/EEC of 15 December 1986 relating to a proceeding under Article 85 of the EEC Treaty (IV/31.458 - *X/Open Group*), OJ L35, February 6, 1987, p. 36, requiring that if access is so limited, an exemption under Article 101(3) TFEU may still be available if the results are licensed openly. Access to the decision-making process can slow or distort progress, but skewed access can lead to distortions and inefficiencies in upstream technology competition and downstream implementation competition. Having objective, relevant and proportional rules for access appears to be the best way to strike the balance.

33 See Commissioner Kroes’ OFE Speech, above, footnote 7: “*Allowing companies to sit around a table and agree technical developments for their industry is not something that the competition rules would usually allow. So when it is allowed we have to look carefully at how it is done.*”

34 See R. Weir “The Final OOXML Update”, above, footnote 10, and G. Moody, “Microsoft, OOXML and the ISO: the Response”, July 13, 2009, available at <http://www.linuxjournal.com/content/microsoft-ooxml-and-iso> <http://www.linuxjournal.com/content/microsoft-ooxml-and-iso> . See also *Allied Tube & Conduit Corp. v. Indian Head, Inc.*, 486 U.S. 492 (1988).

35 Cf. Article 101(3) TFEU. Interoperability refers to the ability of information and communication technology systems and the business process they support to exchange data with fidelity and to enable sharing and utilization of information and knowledge. For citizens, interoperability means they can access, provide and utilize government information using the IT solutions of their choice, without being stymied by closed, proprietary hardware or software

d. Open (published, objective, relevant, qualitative, and verifiable) criteria for technology selection. Standard agreements should be based on the relative merits and price of the technologies involved, to the extent possible. Of course, the advantages, performance and costs of technologies cannot always be known before adoption as a standard. But openness will aid a well-informed debate and choice, and minimizes the risk that standard setting is used as a cloak for an inefficient cartel or a tool to distort inter-technology competition

e. No overstandardization. A standard should be no more restrictive than necessary to meet the objective, and should allow maximum consumer choice without lock-in to a single vendor's product. Where possible, design specifications should be avoided to ensure maximum competition within the standard on quality and product differentiation.

f. Open access to the standard. A standard is "open" only if it is well-documented and published, and available for implementation for all interesting parties, members of the standards body and outsiders alike. This has several key components, relevant for IPR Policies of standards bodies:

– Open information on blocking patents. No one wants to expose himself to *ex post* hold-up. This means that to the maximum extent possible, patents, patent applications, and other IPRs that could block implementation should be made known as soon as reasonably possible before the standard is selected, so that informed decisions can be made whether the IPRs are available and at what terms.³⁶

– No unjustified refusal to license. Unjustified actual or constructive refusals to license essential IPRs, or unjustified delays, are inconsistent with open access to a standard. The right to refuse to license or obtain an injunction at will is the very cause of the tragedy of anti-commons, and therefore should be limited to situations where a refusal is necessary to prevent the opposite (the tragedy of commons, discouraging investment in R&D).³⁷ A refusal or injunction is justified, in other words, if licensee refuses in turn to license essential IPR on FRAND terms, or where the licensee cannot pay or refuses to pay a FRAND rate. For the same reason, termination of a license should not be allowed at will, but only for (actual or anticipatory) material breach that cannot be remedied, or if the licensee refuses to license its essential IPR ("defensive suspension"). Finally, there should be no constructive refusal to license, for instance, by demands for treble damages or imposition of excessive fees.³⁸ An injunction is not justified if the purpose is to magnify the

solutions that do not support open standards. See IBM paper "Open Standards, Open Source, Interoperability and Government Policy", above, footnote 31.

36 See *Rambus*, above, footnote 11.

37 It has been argued that injunctions should be allowed in standards context subject only to the criteria of abusive litigation, which would be the case if the proceedings cannot be regarded as an attempt to enforce legitimate rights but only serve to harass and if they are part of a framework of a plan to eliminate the competition (*ITT/Promedia*, [1998] ECR II-2937). But that ignores the crucial element distinguishing standard setting from a normal situation, namely, that the IPR owners have promised to license on FRAND terms, the standards organization has relied on it leading to an agreement to limit inter-technology competition that would otherwise have existed, and the industry has relied on it by making investments in innovation. Having made such a promise and obtained monopoly as a result, it should be an abuse of dominance to seek injunctive relief to extract royalties higher than those that would have pertained in *ex ante* inter-technology competition.

38 Cf. *Orange Book* case (German Supreme Court), judgment of May 6, 2009, KZR 39/06, on appeal from Court of Appeal Karlsruhe, Case 6 U 174/02, *Orange Book-Standard*. See also judgment of the Regional Court of Düsseldorf of Feb. 13, 2007 in Case 4a O 24/05, *Siemens v Amoi (Zeitlagen-multiplexverfahren)*. Compare also Judgment of the District Court Düsseldorf, Case 4b O 346/05, *Video Signal Encoding*. German Federal Court of Justice, decision of 13/7/2004 - *Standard-Spundfass II*, WuW DE-R 1329, GRUR 2004, 966.

bargaining power of the patentee and expropriate all or almost all profits that the defendant made or expects to make from a complex product implementing a standard.³⁹ This raises the thorny question of fair pricing.

– Fair pricing. No one should be forced to buy a pig in a poke or expose himself to *ex post* blackmailing. Commissioner Kroes confirmed in her OFE Speech that “*rates [must be] fair, and [...] based on the inherent value of the interoperability information (rather than the information's value as a gatekeeper)*” and “*we step in when companies rip-off consumers, in one way or another.*”⁴⁰ Under Article 102(a) TFEU (ex 82 EC), indeed, dominant firms are prohibited from “*directly or indirectly imposing unfair purchase or selling prices or other unfair trading conditions.*” This covers using market power derived from having been included in a standard, to charge excessive prices or impose unjustifiably onerous or unfair terms. The European Court of Justice has confirmed that it may be a violation of Article 102 TFEU (ex 82 EC) for an undertaking in a dominant position to charge a price that is excessive in relation to the economic value of the service provided or the good supplied.⁴¹ In *Port of Helsingborg*, the European Commission confirmed the “value” criterion and used various proxies to conclude that no violation had occurred in that particular case.⁴² To determine the inherent value, and prevent disputes as much as possible, competition law now allows *ex ante* open disclosures of prices and license terms, and even technology auctions.⁴³ In complex standards, however, this is unfortunately often infeasible, because standards take several years to develop, with a sequence of selection decisions. Once a process is locked into a particular direction, technology choice for subsequent development is restricted. Initial experience with *ex ante* declarations of terms indicates that IPR owners have an incentive (a) to delay disclosures of their patents and the license terms until they have achieved a blocking position, (b) buy, swap or develop blocking patents for each alternative in order to prevent real inter-technology competition,⁴⁴ and (c) once they have achieved a blocking position, rush to the table to claim the highest fee in an attempt to pre-empt other IPR owners’ claims. In practice, therefore, there appear to be only two solutions to ensure

39 C. Shapiro, “Injunctions, Hold-Up, and Patent Royalties” (August 2006), available at <http://faculty.haas.berkeley.edu/shapiro/royalties.pdf> (“*patentees whose inventions are only one component of a larger product are systematically overcompensated. The reasonable-royalty floor for patent damages is designed to compensate a patent owner for losses it sustained as a result of infringement, not to punish or deter infringement or even to deprive an efficient infringer of all of the profits from that infringement. But the way reasonable royalties are calculated, particularly for component inventions, has made them into a tool for patentees to capture more than their fair share of a defendant's profit margins. [...] damages reform must be coupled with a solution to the holdup problems created by injunctions. [...] holdup problems in patent cases can be quite significant, but that a relatively simple step—a stay of injunctive relief sufficient to allow the infringer to design around the patent if it can in cases involving reasonable royalties but not lost profits—would significantly reduce that problem*”). See also J Farrell, J Hayes, C Shapiro, and T Sullivan, “Standard Setting, Patents and Hold-Up”, (2007) *Antitrust Law Journal* 74(3) 638; M. Lemley and C. Shapiro, “Patent Holdup and Royalty Stacking”, (2007) *Texas Law Review* Vol. 85:1991-2049, at , <http://faculty.haas.berkeley.edu/SHAPIRO/stacking.pdf> .

40 “Competition and Consumers in the 21st century”, SPEECH/09/486 by Commissioner Kroes, October 21, 2009.

41 See *General Motors v. Commission*, [1975] ECR 1367, and *United Brands v. Commission*, [1978] ECR 207.

42 Case COMP/A.36.568/D3, *Scandlines Sverige AB v. Port of Helsingborg*, Commission Decision of 23 July 2004. See also M. Glader and S. Chabert Larsen, “Excessive Pricing and Article 82”, *Competition Law Insight*, July 2006, at 3-5.

43 OFE Speech: “*If we are to include proprietary technology in a standard, then ex ante disclosure [of essential patents and maximum royalty rates] may help those involved make a properly informed decision. Competition law should not stand in the way.*”

44 Competition law may be powerless to block strategic patent acquisitions or swap arrangements of this kind, if the transactions do not meet the turnover thresholds for merger control. Qualcomm’s acquisition of Flarion’s IPR portfolio is an example.

open access: (1) a clear policy of avoiding all patents or insisting on royalty-free access, as W3C and the open source community do, or (2) a clear and enforceable policy of fair, and reasonable pricing.

The former option – a policy of avoiding all patents that are not available on royalty-free terms without restriction – is adequate and pro-competitive so long as it does not prejudice incentives to innovate. New revenue models suggest, in fact, that at least in the software sector, IPR protection is not the only model encouraging innovation. The W3C's Internet open standards are IPR-free or royalty-free, and are arguably one of the greatest platforms for innovation that the world has seen. Open source software development is encouraged by the prospect of revenues from upgrades, services and complementary products rather than on royalty income (although it depends on the existence of copyright to ensure that open source license conditions are passed on).⁴⁵ Innovative advertising-funded or transaction-funded IT services do not rely on fees from users, and in two-sided markets, giving away one product for free may generate demand for another fee-paying product. The development of free APIs and free interoperability information for a software platform pays for itself because it makes the platform more attractive as additional complementary products become available for it.⁴⁶ The cost of bringing software to market is less than for tangible products. All of these factors are especially pertinent in industries where network effects are strong, because success feeds on itself and magnifies the potential income from these alternative revenue models. These examples from the software area suggest that software-to-software interoperability standards can and should be royalty-free.⁴⁷

In other areas, such as mobile telecommunications networks and computer hardware, the situation appears different. No open source movement exists for telecommunications network technology. In these areas, mandating royalty-free licensing would likely recreate a tragedy of commons and discourage innovation, while allowing IPR owners to charge at will could create a tragedy of anti-commons. To strike the right balance, therefore, a contract of mutual restraint is necessary. This was the intent of the IPR Rules adopted by ETSI in the 1990s, which called for essential IPR owners to commit (before a standard is finalized, at a time that inter-technology and inter-standard competition is still viable) to charge "fair and reasonable" royalties.⁴⁸ This is also mandated by

45 See Y. Benkler's brilliant "Coase's Penguin, or, Linux and The Nature of the Firm", 112 *Yale L.J.* 369 (2002) http://www.yalelawjournal.org/112/3/369_yochai_benkler.html http://www.yalelawjournal.org/112/3/369_yochai_benkler.html.

46 "The return is NOT necessarily about royalties. The return may be that a product that includes the standard as part of it will do better in the marketplace because of the broader adoption of the baseline technology as a standard. The return might be in improved interoperability of a given product or service due to the adoption of that standard. There are competitive reasons for contributions – the hope may be to displace a competitor who is using a non-standardized solution" (J. Matusow, "Balance of Contributors & Implementers", August 2, 2009, <http://blogs.msdn.com/jasonmatusow/archive/2009/08/02/balance-of-contributors-implementers-a-blog-answer-to-rick-jelliffe-s-post.aspx> <http://blogs.msdn.com/jasonmatusow/archive/2009/08/02/balance-of-contributors-implementers-a-blog-answer-to-rick-jelliffe-s-post.aspx>). Similarly, the Apple iPhone truly took off when Apple opened up its APIs and applications became available.

47 See also European Commission, White Paper, Modernising ICT Standardisation in the EU - The Way Forward, COM(2009) 324 final, July 3, 2009, p. 8-9, which recognizes the distinction between the software interoperability and the telecom network sector.

48 See ETSI IPR Rules and ETSI Report "The Way Forward for IPR", above. That the intent was a call for mutual restraint in royalty setting is confirmed by various industry statements, including "NTT DoCoMo, Nokia, Siemens and Japanese manufacturers Reach a Mutual Understanding to Support Modest Royalty Rates for the WCDMA Technology Worldwide", 6 November 2002, available at <http://www.umts-forum.org/http://www.umts-forum.org/> and "Wireless Industry Leaders Commit to Framework for LTE Technology IPR Licensing", statement of 14 April 2008

Article 101(3) TFEU (ex Art. 81(3) EC): in exchange for being allowed to restrict inter-technology competition by agreeing to choose one technology for a standard and exclude others, the conditions for exemption must be met: competition in the products implementing the standard must not be eliminated, and consumers must get a fair share of the benefit. It is also mandated by Article 102(a) and (c) TFEU (ex Art 82(a) and (c) EC) which prohibit a dominant firm (the owner of *ex post* essential patents) from imposing unfair and discriminatory terms, especially where licensees made investments in legitimate reliance on the FRAND promise.

Unfortunately, a contract of mutual restraint can exist only if and so long as everyone plays by the rules. If one patentee breaks ranks and charges the highest royalty it can get away with, would the others grin and bear it, and lower their fees to absorb the price increase? It has been suggested that this would in fact happen, even that it would be “fair and reasonable” and consistent with competition law and FRAND promises for a non-vertically-integrated licensor to extract an amount close to full monopoly rent for a patent, leaving the crumbs for the other licensors and licensees. Consumers will not suffer, the argument goes, since a rational analysis of an “ultimatum game” indicates that (a) licensors of complementary essential patents will restrain themselves and seek only the difference between the royalty charged by the first licensor and the monopoly rent, so as to avoid a “Cournot complements” problem, and (b) licensed manufacturers will maintain the price for the end product and lower downstream profits (reducing their reward for innovation and risk downstream).

This argument is probably correct in cases where two cumulative conditions are met: (a) the patents were *ex ante* essential, absent viable alternatives, and (b) no complementary essential patents nor downstream innovation are needed.⁴⁹ If those conditions are met, there was no inter-technology competition to begin with, and the standard agreement is not caught by Article 101(1) TFEU. But if (a) the patents were not *ex ante* essential or (b) implementation requires complementary patents from other licensors, there are several reasons why this cynical “first mover takes all” approach will not be “fair and reasonable” – apart from the consideration that it

by Alcatel-Lucent, Ericsson, NEC, NextWave Wireless, Nokia, Nokia Siemens Networks and Sony Ericsson, available at <http://www.ericsson.com/ericsson/press/releases/20080414-1209031.shtml><http://www.ericsson.com/ericsson/press/releases/20080414-1209031.shtml>. See also <http://www.nokia.com/A4993368> and http://www.ericsson.com/technology/licensing_programs/index.shtmlhttp://www.ericsson.com/technology/licensing_programs/index.shtml. Cf. also *Siemens v Amoi (Zeitlagenmultiplexverfahren)*, District Court of Dusseldorf, 13 February 2007, 4aO124/05 and *Nokia Corporation v Interdigital Technology Corporation* [2007] EWHC 3077.

49 Cf. *Motorola v Rockwell int'l Corp*, No 95-575-SRL (D.Del 1995). This should be distinguished from the 2004 decision in *Microsoft*, which concerned software interoperability (see fn. 80 above), was a remedy, and where patents were not *ex ante* essential. In that case, the Commission appropriately distinguished between two types of “value” transferred to competitors by the compulsory license that the Commission imposed, in a way that is also relevant to standards cases. It differentiated between (a) “‘strategic value’ stemming from Microsoft’s market power”, and (b) value derived from true innovation. The former is the amount that Microsoft could extract in a hold-up of the users of its interoperability information, considering that the industry cannot avoid that information because of the need for their servers to communicate with Microsoft clients and servers on an equal footing as Microsoft’s servers. The latter is the value derived from true innovation, i.e., the *ex ante* incremental value (if any) over the next best alternative had there been open standardization and an auction before Microsoft became dominant in client PC operating systems and the industry was locked in. See *Microsoft* Commission Decision of Mar. 24, 2004, Case COMP C-3/37.792, *Commission v. Microsoft Corp.*, 2007 OJ L 32, p.23–28, ¶ 1008. The question whether Microsoft’s penultimate royalty offer was “fair and reasonable” was addressed in Decision of the European Commission C(2008) 764 final of 27 February 2008 fixing the definitive amount of the periodic penalty payment imposed on Microsoft. This decision is subject to appeal. Case T-167/08, *Microsoft v. Commission*, OJ C 171/41, July 5, 2008.

will not sound “reasonable” to the proverbial “man in the Clapham Omnibus.”⁵⁰

First, in the EU (and unlike Section 2 of the Sherman Act in the US), Article 102(a) and (c) TFUE prohibit unfair pricing or unjustified discrimination even in the rare case where *no ex ante* competition existed, so long as consumer harm ensues from excessive or discriminatory pricing. A “first mover takes all” approach could cause consumer harm by discouraging cooperation by other patentees and reducing investments by licensees in setting and implementing the standard, while the prospect that this is allowed in standard setting generally would dampen the incentives for dynamic competition by developing new technologies that could be used for future standard generations. Article 102 TFEU is appropriately applied where excessive or discriminatory pricing discourages standard implementation, investment in R&D for future standards, or future standard setting, or where it results in a “Cournot problem” or a “game of chicken” (see below). In the US, this may be caught by Section 5 of the US FTC Act.

Second, in game theory, it may indeed be a rational response to an “ultimatum game” for co-licensors and licensees to lower their revenue expectations to ensure that the total royalty stack does not exceed monopoly rent. But this is the kind of theory that earns some economists the definition of “academics who tell you why what happens in market reality is impossible in theory.” In commercial reality, co-licensors cannot estimate what the monopoly rent is, and will likely try (especially if they have not sunk investments in implementation of the standard) to capture more than the crumbs left by the cynical first-moving licensor. They will want equal or greater remuneration if their technical contribution to the standard was equal or greater as the first mover’s. They can do that by raising their rates to the same level as the first licensor, to try and force his rate down – what game theorists call a “game of chicken”. They need not even raise rates, but could, for instance, spin off part of their patent portfolio to create one or more additional licensors charging equivalent or higher rates for their “new” portfolio. As the *IPCom* and *N-Data* cases indicate, any FRAND promise travels with the spun-off patents, but if a FRAND duty were in fact interpreted to leave freedom to charge whatever the market can bear, as opposed to an amount proportionate to the “value” or total technical contribution to the standard, a FRAND promise is no constraint. The result is a mutual hold-up or a prohibitively high royalty stack, all or part of which will be passed on to consumers,⁵¹ and possibly even failure of the standard.

50 In an “ultimatum game”, one person is asked to share a stack of money with an unknown counterpart, who can react either by agreeing (in which case the counterpart receives what was offered, and the offeror keeps the rest) or by vetoing (in which case neither party receives anything). The “subgame perfect Nash equilibrium” (the optimal rational outcome) is that the first moving licensor (the offeror) receives close to monopoly rent and the remaining licensors and licensees (the offerees) receive just enough not to turn them away from licensing and implementing. Experimental game theory indicates, however, that most people do *not* consider the purely rational optimal outcome “fair and reasonable”. Ordinary individuals playing ultimatum games tend to share more than the rational minimum with their counterpart. This is known as “iniquity aversion.” Cf. for instance A. A. Stanton, “Evolving Economics: A Synthesis”, April 26, 2006, available at <http://mpra.ub.uni-muenchen.de/2369/http://mpra.ub.uni-muenchen.de/2369/>. These studies are relevant, because they give an indication of what reasonable participants in standards bodies (who are not economists, but normal individuals) in fact expect from each other when agreeing on FRAND licensing. See also “Standards, IP and Competition: De Aequitate Non Est Disputandum?”, Helsinki, October 7, 2009, available at <http://www.iprinfo.com/tiedostot/Dolmans.pdfhttp://www.iprinfo.com/tiedostot/Dolmans.pdf>.

51 If all licensees face higher royalties, all would pass them on 100% to consumers. Even if only patent-poor licensees paid high royalties (with patent-rich licensees negotiating a royalty reduction for a cross-license), they would still have the incentive to pass these on to consumers, and patent-rich licensees would likely respond by raising their prices as they face less competitive pressure. Economic analysis indicates, therefore, that consumers suffer either way.

Economic analysis provides a framework for analysis to avoid this tragedy of anti-commons, by defining a “fair and reasonable” royalty not as the rate that the market can bear *ex post* or that the first mover demands, but as the lower of (a) the rate that the IPR owner could have obtained in an *ex ante* inter-technology auction, with different technologies competing for the standard, before the investments are finalized (ignoring any anti-competitive actions or patent acquisitions by the IP owner that diminish *ex ante* inter-technology competition),⁵² and (b) if the IPR owner had an *ex ante* blocking patent, a share of the royalties that is proportionate to the technical contribution the IPR owner made to the standard compared to that of other essential patent owners and taking into account the investments made and risks borne by the licensees. As Commissioner Kroes said in her OFE Speech in 2008: “*I fail to see the interest of customers in including proprietary technology in standards when there are no clear and demonstrable benefits over non-proprietary alternatives.*” The logical extension of that argument is that if there are clear and demonstrable benefits, the total royalty should be no more than the value of these benefits, *i.e.*, the value that licensees can derive from using the selected technology over and above the value they could have derived from the next best alternative. If the necessary information is not available, economists can do a “Shapley value” analysis, or use proxies, such as:⁵³

- i. A comparison with royalties and terms that *other* owners of essential patents reading on the same standard charge for their complementary patents (“proportionality analysis”);
- ii. A comparison with royalties and terms that the patent owner itself charges for *other*, comparable, technologies (“proxy analysis”);
- iii. The “Goldscheider analysis” (which is, however, controversial), which suggests that IP owners in the aggregate should generally be entitled to about 25 percent of the downstream gross profits made on the licensed product.⁵⁴

52 D Swanson and W Baumol, “Reasonable and Non-discriminatory (RAND) Royalties, Standards Selection, and Control of Market Power” (2005) 73 *Antitrust Law Journal* 7. Quoting Swanson and Baumol, the US FTC held in *Rambus* that a reasonable royalty “*is or approximates the outcome of an auction-like process appropriately designed to take lawful advantage of the state of competition existing ex ante [...] between and among available IP options.*” In *the Matter of Rambus Inc.*, FTC Docket No. 9302, Opinion of the Commission on Remedy, February 5, 2007, at 17. For further refinements, see also S. Besen and R Levinson, “Standards, Intellectual Property Disclosure, and Patent Royalties After Rambus”, 10 *N.C. J.L. & Tech.* 233 (2009), available at <http://cite.ncjolt.org/10NCJLTech233http://cite.ncjolt.org/10NCJLTech233>.

53 See *United Brands v. Commission*, [1978] ECR 207 and subsequent cases on excessive pricing. A Shapley value analysis describes a way to fairly allocate gains derived from cooperation among several actors. See http://en.wikipedia.org/wiki/Shapley_valuehttp://en.wikipedia.org/wiki/Shapley_value. Honesty dictates that I admit to not fully grasping the mathematics, but the upshot is that each player obtains a share of the gains that is roughly proportionate to the relative value of his or her contribution. If one player has found a right hand glove and a second player has found a left hand glove, and the goal is to create a pair that can be sold for 6 Euro, both share the revenues 50/50. If two players have found right hand gloves and a third player has found a left hand glove, and the goal is to create a pair that can be sold for 6 Euro, the third player gets 2/3 of the revenues (4 Euro), whereas the first or second player receive 1/6 (1 Euro). (The consumer presumably receives a 1 Euro discount, benefiting from competition between players 1 and 2.) In an ultimatum game, the hard-nosed optimal rational outcome would be for the third player to offer marginal cost + 1 cent to each of players 1 and 2, allowing the third player to keep 5.99 Euro for himself (since they found the gloves and have no marginal costs). It’s rational, but is it fair? Interestingly, it appears that the outcome of a one-shot experimental ultimatum game played by Western players would in most cases also result in player 3 receiving 2/3 (4 Euro) and leaving 1/3 (2 Euro) to player 1 and/or 2.

54 R. Goldscheider, *New Companion to Licensing Negotiations: Licensing Law Handbook* ¶ 7.02[8][b] (2003–2004 ed.). In determining the final percentage, adjustments should also be made for the enforceability and essentiality of the patents, the geographic scope of various patents and their remaining life, the costs of complementary technology needed, the value conveyed by the patents compared to the next best *ex ante* alternative, the risk borne and investments

Proxies are imperfect, but if an IP owner considers that its patents are worth more than a proportionality or proxy analysis suggests, it can (and bears the burden to) prove that its patents are less vulnerable to challenge, have broader geographic scope or a longer life, convey more value compared to the next best *ex ante* alternative, etc than the other essential patents, or that it bore greater risk than usual compared to licensees. There is precedent for this switch in the burden of proof.⁵⁵

– Non-discrimination and no restriction of competition. Participants in standards bodies cannot reasonably be expected to agree to a standard including patents to which they have no access on the same basic terms as other companies implementing the standard. Differential treatment without objective and proportionate justification tilts the competitive playing field and thus prejudices open access. This reduces efficiencies and distorts competition between downstream players to find the optimal implementation. It is, moreover, a way to circumvent royalty constraints, allowing the patent owner to extract monopoly rent by monopolizing the downstream market for products implementing the standard. It is, in fact, a more effective way to do this, since it is difficult to change contractually fixed royalty rates, whereas it is easy for a monopolist to adjust prices for implementations to maximize profits. Monopolizing the downstream market creates the additional problem that it allows the monopolist the ability to manipulate supplies in order to put pressure on members of standards bodies who might otherwise have moved to avoid the monopolist's patents when setting the next standard.⁵⁶

For these reasons, terms and conditions or price squeezes that have the object or effect of restricting downstream competition, or differential treatment based on whether licensee purchases the licensor's downstream product, should not be allowed under Articles 101(3) and 102 TFEU.⁵⁷ Nor should IPR owners be permitted to extract inadequately remunerated cross-licenses, which reduce incentives to innovate in standards implementation and inter-standard competition.⁵⁸ Especially in the case of manufacturers controlling large market shares or in the case of *de facto* or *de jure*-mandatory standards, finally, standardization must not be exclusive and must not prevent the use of additional technology, or the development of competing standards. In the software-to-software interoperability area, where open source is a driver for innovation, this means that open standards licensing policies (to the extent they allow inclusion of patents in standards) should be

made by the licensee relative to the costs and risks borne by the licensor, the volume of sales expected in the market, and so forth. It should be adjusted downwards for instance, in situations where the licensees take more than the usual risk, or where there were adequate alternatives for the patents in question.

55 Cf Case C-395/87 *Ministère Public v Tournier* [1989] ECR 2521, [1991] 4 CMLR 248, para 38.

56 See M. Dolmans, "Standard Setting – The Interplay with IP and Competition Laws – How to avoid false FRANDs", 2008 Fordham IPR Conference, in Hugh C. Hansen (ed.), *Intellectual Property Law and Policy*, Volume 12 (forthcoming).

57 See also Korean FTC Press Release "KFTC took corrective measures against Qualcomm for abusing its monopoly market status in modem chip market; Imposition of fine and issuance of corrective order for discriminative royalty rates, conditional rebates, etc", July 23, 2009. The EC Commission's case-law and practice in the context of essential facilities suggests that licensors of patents that are essential for compliance with a *de jure* or *de facto* mandatory standard should ensure separate accounting for their downstream manufacturing of standard-compliant products, so as to be able to demonstrate that they do not give competitive advantages to their own manufacturing divisions that they withhold from outsiders. The usual objection under US law against price squeeze analysis (that if the supplier has no duty to supply, it cannot have a duty to avoid price squeezing) does not apply where the IP owner promised to license on RAND terms.

58 See also Japan FTC *Cease and Desist Order Against Qualcomm*, September 20, 2009 (on appeal) available at <http://www.jftc.go.jp/e-page/pressreleases/2009/September/090930.pdf> <http://www.jftc.go.jp/e-page/pressreleases/2009/September/090930.pdf>.

open source compatible or at least not discriminate against open source.

To summarize, “*The acid test for an open standard is whether or not it actually permits substitutability and choice among independent, multi-vendor implementations on different technology platforms with acceptable levels of functionality. The diversity of competing applications that support the standard is also an indication of its openness.*”⁵⁹ And under Article 101(3) TFEU (ex 81(3) EC), the license arrangements should be “*allowing consumers a fair share of the resulting benefit*” and “*not [...] afford [...] undertakings the possibility of eliminating competition.*”

It is interesting to see how this call for open standards fits in upcoming Commission policy in the area of ICT. RAND Europe recently published a report for DG Information Society, entitled “*Trends in Connectivity Technologies and their Socio-Economic Impacts – Policy Options for the Ubiquitous Internet Society*” (the “RAND Report”).⁶⁰ Behind this somewhat impenetrable title are 200 densely written pages with a complicated – but quite interesting – analysis that may well serve as the foundation for DG Information Society’s ICT Policy in the next decade. The RAND Report identifies a number of current technical and economic trends, which could lead to three possible scenarios for what the Internet may look like by 2020, reflecting distinct combinations of public vs. private governance, open vs. closed technologies, and competitive vs. collusive markets, including the following two extremes, whose description is revealing in itself:⁶¹

– The *Scattered World* scenario (the Internet in 2020 as collection of competing networks based on closed technology) reflects a “*future of cutthroat monopolistic competition, unrestrained by active and effective antitrust and other regulation [...]. The fragmentation of competition and low levels of vertical and horizontal integration have as a counterpart low levels of inclusion and worrying levels of inequality.*”

– The *Connected World* scenario (the Internet in 2020 as a network based on open technology, driven by public investments and collaboration between firms) paints a “*future where companies collaborate [...] [and] governments [...] take a cooperative lead in setting rules to optimise global public value creation [...]. [F]irms have to compete (and make their money) on the merits of what they provide rather than the ability to exclude rivals. [...] [I]nteroperability is a powerful public good, and governments are particularly vigilant against the risk of foreclosure by «bottleneck» firms or proprietary standards, using antitrust regulation, support for open standards and targeted public procurement to ensure a sustainably level playing field with high quality of service and reasonable prices. A potential limiting factor is that the speed of innovation [...] is slowed by the natural pace of government initiatives [...]. This world is very inclusive, including excellent technologies to assist those that need assistance to participate.*”

For each scenario, RAND analyzes the expected status of innovation, consumer choice, privacy, social cohesion and equality, and identifies a number of critical problems, positive developments,

⁵⁹ See IBM paper “Open Standards, Open Source, Interoperability and Government Policy”, above, footnote 31.

⁶⁰ Study conducted for DG Information Society and Media, July 2009, available at http://ec.europa.eu/information_society/activities/foi/library/docs/final-report-nosec-clean.pdf http://ec.europa.eu/information_society/activities/foi/library/docs/final-report-nosec-clean.pdf (“RAND Report”).

⁶¹ RAND Report, above, footnote 60, p. xix.

and uncertainties. Experts were then asked to look back from the future and identify the key policy choices that will create desirable outcomes and those that create problems. The report discusses IPRs, privacy, data protection, infrastructure investment, e-commerce and other policy issues. Interesting conclusions are also drawn with respect to open standards and interoperability and net neutrality. RAND mentions, for instance that *“The interconnectedness of the [Internet] challenges competition as the sustainable engine of continual improvement. Network externalities favour ‘tipping’ into monopoly and competition weakens”* and the report worries that this may give rise to *“a desire to limit interoperability [more] than a desire to innovate and offer effective choice.”*

RAND concludes (p. 132) that Europe and most other major jurisdictions have not tried to regulate the Internet, but *“as its spread and importance increase, this may no longer be possible, especially as other regulated activities ‘escape’ on-line and new policy concerns emerge.”* The report warns that alternatives to regulation should be considered early in the policy process. For instance (p. xxvi), *“the EC can encourage efficient competition among technologies and discourage inefficiently-high incompatibility, through creation or coordination of multi-stakeholder platforms and networks, and by applying multi-stakeholder governance principle. These would be enabling the adoption of common standards and market wide approaches to public policy concerns.”* Open standards as defined above would appear to present a perfect mix of flexible multi-stakeholder arrangements, ensuring an adequate balance between the need to foster private sector innovation and the need to avoid technological lock-in or gridlock. Indeed, the RAND Report (p. 145) identifies a dozen or so of key goals for DG Information Society to pursue, which include:

- *“Guarding openness and open networks”*;
- *“Champion common standards and pre-competitive collaboration”*;
- *“Champion interoperability in all its forms”*

Drawing attention to economic analysis and using existing literature, the RAND report identifies various tools and associated challenges to achieve these goals (p. 141 ff), which it is worthwhile to mention:

- RAND proposes to use a range of *ex ante* and *ex post* regulation such as spectrum allocation, competition regulation, telecommunications pricing, interconnection, content regulation, fair competition and merger regulation, consumer protection, privacy, etc. At the same time, it identifies as a key challenge *“to balance lightness of touch with credible effectiveness, [...] and to prevent capture and/or foreclosure that distort markets and the development of the Internet.”* Following the principles of open standards set out above should go a long way to meet this challenge, by maintaining adequate involvement of the private sector, while preventing capture and foreclosure.
- RAND approves of EC involvement in IPR regulation, to provide fair returns on risky inventive activity and as a market-based tool to signal where ideas are best applied. Interestingly, the report warns about the *“the one-size-fits-all nature of the most common forms of IPR protection”*. The open standards approach described above is useful to avoid the problems of this “one-size-fits-all” IPR protection, for instance, by allowing for a royalty-free standards approach for software interoperability and royalty-bearing standards in telecommunications and hardware. Other key

challenges RAND mentions are “*the potential for failure in the market for IPR, the possibility that market power in the market for innovation will spill over into markets for goods and services or vice versa and the possibility that predatory use of IPR [...] and strategic incompatibility may undercut the hoped-for benefits.*” Again, the open standards framework described above would address these concerns, by encouraging standard setting to address incompatibility failures, and ensuring that the necessary patents are available on FRAND terms.

– RAND encourages standard-setting and support for standard-compliant products, including by thoughtful procurement policies in favour of open standards. RAND adds: “*Key challenges here are to maintain openness of standards (to avoid lending public support to proprietary standards), to balance the interoperability advantages of standardisation against the potential loss of diversity and inhibition of innovation and to ensure that standardisation enhances the innovativeness and competitiveness of the European economy.*” While the RAND Report does not further define what “openness of standards” means, and does not propose ways to balance the need for interoperability against the need to maintain product diversity and innovation, it is submitted that the open standards principles mentioned above provide the solution that fits perfectly.

In one respect, however, there is a curious point in RAND’s analysis, concerning the role of competition law to guarantee a system of open standards that maintains innovation while preventing IPR gridlock. The RAND Report (p. 131-132) worries about “*the tendency of Internet markets to tip into monopoly*” but then warns:

“Anticompetitive behaviour can’t always be detected or prohibited ex ante, but ex post remedies (after lock-in has occurred) may be too late, and there may be no counterfactual evidence to demonstrate that alternatives are viable if lock-in is widespread. Moreover, many of the specific activities that firms might use for predatory purposes (e.g. proprietary standards, low “penetration” pricing, etc.) are also essential in order to attract complementary content and services to Internet platforms capable of providing effective competition. Therefore, conventional antitrust policy may be less effective than consumer protection policy or supporting activities that enable users to coordinate moves to superior entrants, and participatory self-regulation may be more effective than IPR policy in deterring or overturning “stealth patents” in public standards.”

It is certainly legitimate to ask whether sector specific regulation should trump general antitrust regulation. With all due respect for the RAND Report, however, the conclusion in this paragraph and some of the thinking in it are at first sight hard to grasp.

First, what exactly are “*activities that enable users to coordinate moves to superior entrants*”? Does RAND propose that the Commission should itself compare different technologies and organize users to move towards solutions that the Commission thinks are superior? RAND recommends that the information needed to allow users to move to other platforms should be made available as a matter of public requirement, or made available by regulatory authorities themselves.⁶² This is consistent with the openness principles set out above. But RAND apparently goes further: where the mere provision of information is insufficient, public authorities might even

⁶² Jonathan Cave kindly explained this and pointed out that Ofcom has undertaken this in its efforts to overcome lock-in among ADSL subscribers unable to obtain MAC codes or even to make authoritative and meaningful comparisons of Quality of Service and related attributes of IPSs’ offerings.

coordinate or encourage moves to alternative platforms. It is submitted that where possible, this should be left to the market, within the parameters of competition law and open standards. Authorities should encourage moves to alternative platforms only where markets clearly fail, and where it is objectively undisputable that the alternative platform is superior from a public interest perspective. In that context, it is not clear what legal basis RAND proposes the Commission should rely upon if not on the competition rules and a public procurement policy favouring open standards.⁶³

Second, it is true that *ex post* remedies may not *always* be effective and anticompetitive behaviour may not *always* be detected or prohibited *ex ante*, but where it can, why not use competition law? As the guardian of the Treaty, the Commission should not be allowed to disregard distortions in a sector as important as the Internet. Lessons can be learned from past failures in remedies – as the Commission did in *Microsoft*, where the 2009 Browser Choice Commitment and Interoperability Undertaking are much more promising than the remedies in the 2004 Decision. Indeed, enforcement action can be a remedy in itself where it deters future violations by the firm or standards body in question, and by others. And in those cases where remedies and anticompetitive behaviour *cannot* be detected or prohibited *ex ante*, how could the Commission possibly devise a consumer protection policy that is more effective than competition law? Even if the Commission has a crystal ball and is able to define appropriate *ex ante* rules, these could more easily and more quickly be set out in the Commission's Guidelines on Horizontal Agreements and applied in cases.⁶⁴ The most effective solution, it is submitted, is to define "open standards" as a condition for exemption of standards agreements under Articles 101(3) and 102 TFEU. This in fact provides some kind of *ex ante* remedy: if these "open standards" conditions are applied during the standards process, the outcome of the standards process is likely to be consistent with competition policy, obviating the need for *ex post* intervention.

Third, it should be recalled that the conception and growth of the Internet was government-funded and took place in a public sector and university environment. As the private sector takes over, and the risk of lock and hold-up emerges, it becomes *more*, not less, important to apply the principles of competition law.

Competition law has the merit of providing a flexible framework that does not require adoption of additional rules. After the closure of the *Qualcomm* case without action in the EU (although action was taken in Japan and Korea and some settlements were reached), some might question whether competition law can be used to prevent consumer exploitation. Lemley argues that it is even

63 See also European Commission, White Paper, Modernising ICT Standardisation in the EU - The Way Forward, COM(2009) 324 final, July 3, 2009, p. 6-7. The requirement in procurement rules, that requests for proposals should refer to European standards where available, serves to ensure maximum consumer choice and dynamic competition, and these goals could be subverted if closed or proprietary standards qualified for preferential treatment under procurement rules.

64 It is unclear whether there is a legal basis in consumer protection provisions of the Treaties, and adoption and implementation of appropriate instruments will take years and be controversial, if it can ever be achieved. It may be possible to amend the Standardization Directive to insist on open standards for European standard setting (Directive 98/34/EC of the European Parliament and of the Council of 22 June 1998, laying down a procedure for the provision of information in the field of technical standards and regulations and of rules on Information Society services (OJ L 204 of 21.07.1998), as amended by Directive 98/48/EC (OJ L 217 of 05.08.1998), but even that will be subject to controversy and intense lobbying by firms that benefit from hold-up practices. Fostering open standard setting under Articles 101(3) and 102 TFEU and the Guidelines for Horizontal Agreements seems the most efficient solution, certainly in the short term.

undesirable, and that antitrust is “a back-stop that’s going to apply only if private efforts in SSOs and IP law have already failed us.”⁶⁵ The problem is that private efforts can be blocked by firms that benefit from hold-ups. Experience suggests that this would likely prevent standards organizations – working by consensus – from modifying their IPR Policies meaningfully. At best, this will take a long time, and at worst, it will never happen satisfactorily. The same firms also argue in court that FRAND promises are unenforceable in contract or even under promissory estoppel principles, and are merely a promise to negotiate. Amendments to IP law are not much more promising either. Getting twenty-seven Member States to change their IP laws to prevent hold-ups is probably a pipe-dream. Member States judges are equally unlikely to change the law, since they are much less policy-oriented than their US brethren. Antitrust law seems to be the only available tool to achieve a focused, timely, EU-wide solution. Regulation and IPR laws are blunt instruments with “one-size-fits-all” impact, whereas competition law allows intervention with surgical precision, permits remedies appropriate to address the precise problem and strike the right balance in the specific circumstances of the case, and creates flexible precedent that can be adjusted to new fact patterns.

More important, the public policy concerns under US law that advocate against use of antitrust law (the heavy burden of treble damages, the extraordinary high cost of defense as a result of extensive discovery, the risk of spurious litigation driven by contingency fee arrangements and class actions tried before juries, the need to prove intent) are absent or much less of a concern in the EU. Moreover, contrary to Section 2 of the US Sherman Act, Articles 102(a) and (c) TFEU prohibits excessive pricing and unjustified discrimination restricting downstream competition, and neither the courts nor the Commission are at liberty to ignore the legislator’s intent in that respect – *a fortiori* in the context of a FRAND promise.

This is not to suggest that the Commission should initiate competition law proceedings under Articles 101 or 102 TFEU against standards bodies with inadequate procedures and poor IPR Policies, on the basis of an argument that standards organizations with inadequate rules do not qualify for exemption under Article 101(3) TFEU. Nor should the Commission take away the special status of CEN, CENELEC or ETSI because their policies are deficient. Taking action against standards bodies, or prohibiting individual standards, is in many cases impractical, such as with respect to international organizations like ISO and IEEE.⁶⁶ In all cases it means punishing the victim rather than the perpetrators of standards manipulation or hold-ups. It is better to take firm action against companies that distort standards practices, impose excessive royalties, or impose restrictive licensing terms.

Fourth, the RAND Report mentions that there may be no counterfactual evidence to demonstrate that alternatives are viable if lock-in is widespread. But competition law provides an elegant way out. In *Rambus*, for instance, there was evidence that Rambus took steps to conceal its patent and patent applications. Had it really thought that its technology was better than the available alternatives that JEDEC (the standards body) considered, why did it take such steps? By concealing the information, it prevented the counterfactual from materializing, and it should bear

65 See M. Lemley, “Ten Things to Do About Patent Holdup of Standards (and One Not to)”, (2007) 48 B.C. L. Rev. 149, 151-55.

66 It is also impractical with respect to ETSI, whose IPR Policy was, after all, granted negative clearance. See OJ 1995 No C 76, p. 5, and 25th Report on Competition Policy 1995, pp. 131-132.

the consequences for that: In cases where IPR owners conceal evidence of IPRs, or impose confidentiality clauses preventing licensees from warning standards bodies that royalties are too high or terms are exclusionary, the burden of proof should be switched to the IPR owner to show that no viable alternative existed for their technology and that the standards body would have included their IPR anyway without a FRAND licensing obligation. Case-law provides precedent for such a switch in the burden of proof.⁶⁷

A final comment concerns the statement that “*many of the specific activities that firms might use for predatory purposes (e.g. proprietary standards, low “penetration” pricing, etc.) are also essential in order to attract complementary content and services to Internet platforms capable of providing effective competition.*” Of course, low pricing or even giving away products or services may be legitimate in order to foster a network effect or attract business in a two-sided market. Similarly, building products based on proprietary technology like Apple’s iPod and iTunes is a legitimate business model. But using closed standards is not “essential” to attract complementary content or services – they are at best neutral in attracting complementary products, and tend to limit competition from substitutes.

To conclude: The RAND Report should be commended for recognizing the importance of open standards, and the criteria suggested above (including the conclusion that software-to-software interoperability standards should be patent or royalty-free where alternative revenue models exist) fit well within this framework. The paragraph on p. 132 should not be relied upon to throw out competition policy as a tool to maintain an open Internet. The comment that “conventional” antitrust policy is less effective, is better understood as a call for application of *more innovative* competition policy to strengthen open standards and foster consumer welfare and consumer choice, which are the objectives of competition policy. This is also consistent with comments elsewhere in the RAND Report (p. 100, emphasis in original):

“One key element is the importance of market competition in motivating and funding the development of innovations and in determining their availability, affordability and the resulting impacts on societal objectives. As a result, ...effective competition policy remains essential. This raises new challenges for existing (technical and economic) regulators in relation to IPR, bundling and the treatment of joint ventures. More profoundly, it can change the synergistic relation that has traditionally existed between competition and consumer protection policies. To avoid capture, unjustified market distortion or an inappropriate balance of efficiency and innovation, it is necessary to ensure that competition policy promotes the efficiency benefits anticipated from competition rather than competition for its own sake.”

The EU may consider legislation that lays out a common set of rules for “fair play” in standards negotiation. But while regulation and IPR laws are blunt instruments, competition law properly and energetically applied allows intervention with surgical precision, permits remedies appropriate to address the precise problem and strike the right balance in the specific circumstances of the case, and creates flexible precedent that can be adjusted to new fact patterns. The Commission showed this when it negotiated a browser choice screen for Windows: Competition in browsers creates opportunities for alternative browsers that comply with open standards such as HTML5, and if enough users exercise that choice, developers will have incentives to use those open

67 Cf Case C-395/87 *Ministère Public v Tournier* [1989] ECR 2521, [1991] 4 CMLR 248, para 38.

standards as well, keeping the Internet open. This remedy, therefore, allows the market to speak. Let's hope that the proposed revision of the Guidelines on Horizontal Agreements will reflect this open standards approach, that Commissioner Almunia will apply it, and that Commissioner Kroes will integrate competition policy when setting the Digital Agenda in her new position.

About the authors

Maurits Dolmans is a partner at Cleary Gottlieb Steen & Hamilton LLP, Brussels. The author has been involved in various cases relating to standards, IPR and competition, including most recently on the side of complainants in Microsoft, Rambus, IPRCom, and Qualcomm, and on the side of defendants in HD-DVD/Blu-ray and pending other matters, and has acted as outside counsel for standards bodies such as ETSI. This paper, however, reflects his personal views in an evolving debate, and do not bind the firm or its clients. The author thanks his colleagues Thomas Graf and Lars-Peter Rudolf, Jonathan Cave of Warwick University, and Hal Varian of UC Berkeley for their insightful comments and critical review.

Carlo Piana is an Italian IT lawyer based in Milan as well as a Free Software and digital liberties advocate. Since 2004 he provides consulting to the FSFE and assists the same in battles for competition and open standards. He has represented FSFE and the Samba Team in the antitrust European litigation for obtaining the full interoperability information of the Windows networking interfaces. He is a member of the Editorial Committee of this Review and a strong believer in Free Software and Digital Human Rights.

Licence and Attribution

This paper was first published in *Published in:*
Concurrences N° 1-2010, n°30204, pp. 13-38
<http://www.concurrences.com/>



It was republished courtesy of the Review Concurrences in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifosslr.org>. The introduction is a work of Carlo Piana, approved by the Author.

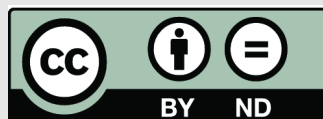
This article should be cited as follows:

Dolmans, Marurits (2010) 'A Tale of Two Tragedies – A plea for open standards, and some comments on the RAND report', *IFOSS L. Rev.*, 2(2), pp 115 – 138
DOI: [10.5033/ifosslr.v2i2.46](https://doi.org/10.5033/ifosslr.v2i2.46)

Copyright © 2010 Maurits Dolmans.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Project Harmony:

Inbound transfer of rights in FOSS Projects

Amanda Brock^a

(a) General Counsel, Canonical Group

DOI: [10.5033/iffosslr.v2i2.49](https://doi.org/10.5033/iffosslr.v2i2.49)

Abstract

Software engineers participating in open source software projects are frequently asked to sign Copyright Contribution Agreements ("CCAs") with respect to the contributions they make to those projects.

As lawyers we are happy to read and understand copyright agreements and licences. As lawyers, few of us can code. Software engineers (at least in Canonical) seem to understand more law than might be expected, but their expertise is software, not law. The proliferation of forms and wording in the CCAs mean that they are routinely asked to sign up to a diverse selection of CCAs differing in meanings and intentions. Valuable coding time is wasted as they wade through legal wording.

This is not only an issue both from a productivity and efficiency perspective, but is also problematic in ensuring that developers understand what rights they grant in their work..

Project Harmony came into being in the Summer of 2010. Its birth followed a long incubation and much discussion within Canonical and with parties such as SFLC and lawyers specialising in FOSS. It is intended in the first instance to tackle CCA Proliferation.

Keywords

Law; information technology; Free and Open Source Software; Copyright; CCAs; Copyright CCAs; Copyright Licence Agreement; Project Harmony

Info

This item is part of the [Articles](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement. This article has been independently peer-reviewed.

Project Harmony – a response to the proliferation of contributions

I joined Canonical in February 2008. Sitting between the developer community and commercialisation of FOSS (Free and Open Source Software), Canonical (through the Linux distribution it represents, Ubuntu) is at the forefront of change and progress in Open Source.

It was a culture that shocked the life back into me after time away from the IT sector. How anyone can fail to love hurtling forward on adrenalin at 100 miles an hour is beyond me. The collaborative nature of the developers has rubbed off on the lawyers involved in FOSS. It is a wonderful experience for a lawyer to find free and open discussion and a collegiate approach between colleagues across companies, countries and firms.

Within a very short time of joining Canonical, with its large developer staff, the piles of paper landing on my desk included CCAs –a consequence of the developers saying: “we have a lawyer now!”

I am not a developer. I had not worked on any other open source project before joining Canonical. I was slightly confused...

An open source software project, whether run by a commercial entity, a not-for-profit or otherwise, frequently establishes a "custodian" of the software. I use the term “Project” to cover both the Custodian and the software.

The custodian is responsible for the wellbeing of the code in the Project, including co-ordinating [release management, feature adoption] maintenance and bug fixes. However, what legal rights that custodian holds in the code and how the custodian manages and administers the legal rights and IP varies between Projects. In most cases, the Project receives contributions from individuals and those individuals form part of a community. The formation of a community is not obligatory and is by no means universal: there are projects which involve the commercial development of FOSS code (which, in development terms, are frequently indistinguishable from proprietary developments), which are internally sponsored by a commercial entity and which do not develop an external community. However, the establishment of an enthusiastic community is a defining characteristic of successful FOSS Projects.

The CCAs I was seeing as Canonical's first lawyer were from many different Projects, some commercial, some not. There were all sorts of CCA documents - copyright assignment agreements (including some misleadingly entitled copyright licence agreements); agreements containing tough obligations on contributors providing free code without payment for their work; agreements lacking clarity as to who contributed as an individual and where a corporate entity should sign for the contribution, and all sorts of wording which had clearly been edited by well-intentioned non-lawyers (NALs). The lack of clarity can even cover issues as fundamental as what work is being contributed, what the CCAs are meant to cover and who the licence or assignment is being given to (for example, where no formal legal entity has been created for a Project).

The outbound licences on which FOSS Projects distribute their software to Projects downstream and other users, are, by comparison, very neat and structured. A couple of decades of active licence creation, followed by consolidation in many projects as a result of the efforts of OSI and others means that, in effect, the licences have become regulated. Of course, this regulation is not by any appointed authority, but is an emergent characteristic of industry and community engagement. The organisations that have emerged out of the community provide boundaries and guidelines as well as, in some cases, approvals for these licences. As a new FOSS lawyer, I was learning fast about the work done by OSI and others in reducing licence proliferation in outbound FOSS distribution.

Many involved with this Journal are much more qualified to write about these licences than I and I won't go into more detail here. But, it struck me as odd that where industry activity had led to FOSS licences becoming clearer and more consistent, in contrast, there was no similar activity in relation to this mass of confusing, potentially onerous and sometimes incorrect CCA documentation that I was reading.

As lawyers we know that there are very few judicial decisions on software licensing in general and, more specifically, free and open source software. We realise that we are operating internationally, without precedent and in many jurisdictions. The paths we tread with respect to these licences are carefully planned and whilst we may not have legal authority to rely on and use as a basis in making our decisions, one thing our industry is not short of is opinion.

Whether made by individual lawyers or by developers (in which case it is likely to be prefaced by "IANAL" - I am not a lawyer - and which is inevitably followed with "but...."), we have the thoughts of the great and the good of the FOSS development community and its lawyers on our outbound licences available to us, through articles, books and blogs. We may not be able to guarantee how a court will interpret a licence provision or its enforceability from one jurisdiction to the next, but we have an idea of why that licence came into being what, the intent of its creator in drafting its provisions are and why someone might choose to use it as opposed to another licence. The same is not true of CCAs.

For some time we discussed this internally in Canonical with both management and developers (and in many cases the two are the same). With the support of Canonical's senior management, I started to speak to my contemporaries and the bodies representing the FOSS Communities, such as SFLC. From these discussions and our recognition of the need to improve CCAs and to remove proliferation of CCAs, Project Harmony was set up.

Project Harmony Terms of Reference

Terms of reference for Project Harmony were circulated after our US kick-off meeting on 16 June, 2010 and agreed by our follow up London meeting in July of that year. They are:

Project Harmony is made up of a group of industry interested parties, from companies, projects and those with personal interests in FOSS. It is open to all interested parties to join. Its initial goals are to avoid proliferation in CCAs across

FOSS software projects where those organisations chose to work with CCAs. In doing this we hope to assist organisations which use CCAs by providing standardised variable templates with clear and concise explanations; to come to a common understanding on these; and to recognise the relative maturity of FOSS by dealing with its internationalisation. Our goal is to make the process of contributing to FOSS projects easier for developers regardless of who their employers are. We believe that standardised CCAs should make it easier for developers to contribute regardless of who their employers are

We may look to establish this forum for these CCAs into an organisation which will have a long term role in administering this area.

To achieve our initial goals we will be working together to plan and to create:

**ContributionAgreements*:*

*A suite of standard language which works across international borders and which provides standardised wording for FOSS contributor agreements, minimising the need for legal review. We understand and accept that there are varying positions and decisions in different projects based on commercial positions, community preferences and the wording needed to meet this. **Inter-CompanyAgreements*:*

In addition it is recognised that there would be an advantage to put in place "inter-company" agreements for employee contributions to other company or projects' software. The intention of such documents is to remove the need for employee signature of project contributor agreements and to equalise the playing field between organisations.

It has been repeatedly emphasized in our meetings that clear and useful FAQs are an essential output of Project Harmony and a goal of the Project.

We are currently working out whether we will deliver a single modular document encompassing options or if we will follow a model more like Creative Commons, multiple documents. We will certainly learn from the past, and whilst I may be relatively new to this type of activity, other participants were involved with Creative Commons and GPLv3's creation.

Some Thoughts on Copyright

I have set out in the next few pages some thoughts around Copyright and CCAs. These are some general thoughts, trying to look factually at various options and copyright issues and are certainly not intended to recommend any one solution over another. They are also not intended to represent the output that we will provide from Project Harmony, but they do encompass some of the considerations we have in the Project.

To the extent that I have failed in my impartiality, quality of explanation or accuracy, the thoughts

that follow are my own and not those of Harmony. One of the best things about the Project Harmony is that we have managed to pull together an expert and considered group of people, who are not only very learned in their fields but who are, some might say, very unharmonious in their beliefs! Some do not see the need for CCAs. Others have been very vocal in their opposition to assignment or to developer warranties. Some think assignment is the only answer and others prefer licences.

There is room in our Project for all views and as can be seen in our terms of reference, to meet those, we need to create (a) document(s) and supporting FAQs which will achieve industry acceptance, to include all views.

Why have a CCA?

In climbing my personal CCA mountain during my first few months at Canonical, it also took me a while to understand why some Projects felt they did not need a signed CCA at all, whilst other Projects would not accept any contribution without signature of their CCA. Surely it was not just differences in their legal advice that made such vast differences between Projects?

Clearly any Project requiring an assignment must have a document. In many jurisdictions an assignment of copyright or other IP rights is not valid if it is not in writing and signed.

I discovered that Projects often have governance provisions in place to regulate participant's conduct and the contribution of code, such as Canonical's Ubuntu Code of Conduct. Participation is subject to adherence to these rules and some projects have historically considered and may well continue to consider that these are enough, through the development of community norms and extra-legal self-enforcement. The Linux kernel is a good example of this and relies on exceptionally strong Project governance.

CCAs may offer some benefits. Initially people may have contributed or contribute under their nickname or without their full identity being known. Their identity may have been known, but if they later became uninterested and moved on, they might also become difficult to contact. CCAs provide a means for a FOSS project to track who is providing code or making contributions into the project and form part of many projects' governance.

They may avoid problems by showing:

- who contributed what
- documenting their agreement as to how their contribution is used
- confirming that the contributor had the right to make the contribution

In short, CCAs can be used to confirm the pedigree of the contribution and in turn, the pedigree of the Project. To grant its outbound licence in the IP in its software, the Project holds itself out as having the right, to grant that licence. So it must either be the copyright owner or have the right to do this under (an) appropriate licence(s) to it. If it does not have that right then it cannot give a valid licence. Each project is responsible to those it distributes to. Such a licence could be unwritten, but that could not in the case of an assignment and even in a licence this pedigree

requirement may not be met.

A CCA will deal with licensing and assignment of rights from a legal perspective, but its also a political document. The CCA reflects the organisation or project's positioning in the market place and the Project's beliefs. It can be very different to look at a CCA to be used by software in a commercial context such as Canonical and in a not for profit such as Apache. These Projects each have differing needs which a CCA has to meet. So, we have to accept right at the start that CCAs need to meet not only legal requirements but also these political and social needs.

A brief discussion of Copyright

The main IP (Intellectual Property) right attaching to code is copyright. In some jurisdictions code can also be protected by patents. I do not personally believe that patents offer a good or helpful protection for code. However, this is not the place to argue those views. Patents exist, and therefore need to be considered by Harmony as part of its process around CCAs (whether we like them or not) and that consideration is referenced later. I am focusing on copyright, therefore, in this piece.

The New York law school article in this issue of IFOSSLR clarifies well the overall issues (at least under US law) faced by Projects and businesses in the FOSS world and highlights how these cascade to the individual engineers. I also set out a few pointers below.

Copyright ownership and Licensing

Where someone creates a copyright work, other than in the course of their employment, that person is the holder of the copyright. If a work is created in the course of employment, then the work and the copyright in it belong to the employer in the UK, in most of the US and many other jurisdictions. CCAs need to be cognizant of this, so that the correct rights are transferred from the correct donor to the correct recipient. Copyright can only be held by a legal entity, that is an individual or a company or other organisation having legal standing. Under English law at least, an assignment to an unincorporated entity is likely to result in some form of trust arrangement, whose structure can only be revealed with certainty by the courts.

For some FOSS projects which have not established a legal entity, it would not be possible for the "Project" to own the copyright except in this uncertain and unsatisfactory way. The original creator of the software may hold the software as an individual. One option is to allow a trusted third party to hold the copyright in way which gives the original creator that the third party will deal with it in an appropriate way. The FSF is an example of an organisation that acts as a trusted third party and has an assignment agreement which allows Projects and their contributors to assign copyright to it.

Where multiple people contribute and create the copyright in the code, then without a copyright assignment (or a common employer) there will be multiple owners of the code. In the UK if each person's contribution to the work is not something that can be separated from the others', then , without any transfer of rights, that results in a work of joint copyright. Where a joint copyright

work exists, all of the owners jointly own the whole and all of the owners would have to consent to any use of their rights in the work. So, if they disagree there will be problems.

On the other hand if each contributor's part can be distinguished, then each person owns their own piece of code and the copyright in it under English law. Whilst each owner can make their own decision about this, co-ordinating the multiplicity of interest in relation to issues like Project licensing can be difficult.

In many ways, shared or joint ownership sounds like the ideal, and in line with the collaborative approach taken by FOSS projects. Shared and joint copyright ownership can be problematic not just in terms of enforcement but practically, in terms of tracking owners of the shared copyright. First, where there is not a single owner of copyright in the whole, there is no single party who is able to grant rights to use the code, in the UK at least except subject to the terms a licence all owners may have entered into.

Secondly in some jurisdictions, to enforce the copyright and bring a claim against someone who infringes that copyright: under US law all the joint owners have to consent. An assignment allows the copyright holders an instrument to collectively defend and enforce their copyright without giving away "the equity ownership", as in many jurisdictions you must actually be the copyright holder and not a licensee to bring an action.

This means that to take these actions the project would have to find all of the contributors and get their agreement. In older FOSS projects where there is no "pedigree" *i.e.* we don't even know who all the contributors are this could be a nightmare. In projects where there is a pedigree, the issue would of course be having all parties come to an agreement.

Where multiple people create or contribute to a copyright work, or code with copyright in, then a single entity or legal person can own that code. For this to happen there has to be a legal assignment of the code from the creator of the code (unless they are an employee when this will be automatic) to the holding entity. The holding entity could be a commercial legal entity, or a trusted third party such as the FSF.

In a CCA with an assignment, the contributor assigns copyright and is simultaneously granted a licence back to allow him or her to use the software, from the Project. If this is very broad then the licence back allows the contributor to retain full rights to re-use, distribute, and continue modifying the contributed code. Apart from allowing a single entity to manage the IP of the whole Project as discussed above, this is almost the same as an exclusive licence.

Where an assignment is used there is a legal requirement that this is in writing and signed.

Contributions under an exclusive licence

Projects may not require assignment, but instead have a CCA which includes a licence. If this licence is an exclusive licence then commercially and practically this is to all intents and purposes the same as an assignment.

A Project which is the recipient of code under a CCA granting an exclusive licence, of course has

to consider possible re-licensing to downstream recipients (or changes in the downstream licence) and enforcement. The re-licensing can be covered by a mechanism contained in the CCA.

However the contributors' rights will to a large extent depend on the licence back given by the Project. Unlike a sole licence, an exclusive licence to a Project will mean that the contributor is not free to use his contribution. The effect of an exclusive licence on the contributor can be as restrictive as an assignment.

CCAs containing a Non-Exclusive Licence

If the licence is non-exclusive, then the usefulness of such a licence to the Project will very much depend on its wording. The rights must be wide enough to allow the Project to use the contribution itself as appropriate, to distribute it under an outbound licence. The CCA should, of course, deal with any outbound commitment in terms of the types of outbound licence under which the Project may employ.

How well this will work practically, commercially and legally will depend entirely on the drafting.

The Project may have difficulty in enforcing the copyright in its work against infringers where it has non-exclusive licences in its CCAs (in the UK, the Copyright Designs and Patents Act 1988 does anticipate that a copyright owner can authorise a non-exclusive licensee to enforce copyright in certain circumstances, but the scope of the provision is unclear and will provide the alleged infringer with scope for argument). If the CCA does not deal adequately with outbound licensing the Project may also find itself unable to release under different or additional licences both now and in the future.

Outbound Licences

One further challenge faced by FOSS projects specifically and which was encountered by Mozilla, when they wished to increase flexibility by enabling their codebase to be licensed under more, and more liberal licence: where a CCA provides that the Project use a specified outbound licence. it may not be possible for the Project to change the licence for a later version or to release under other licences in parallel

It may be that over time or for a particular instance, the Project wants to move to a new or additional licence. If the Project does not own the copyright or have a right in the CCA to do this then unless all of the owners can be found and persuaded to agree, this cannot happen (in the absence of a licence granting that right, or without extracting from the code-base the contributions of errant contributors). Ownership of copyright means that the copyright owner can more easily change the licence under which the software is released. Through a contractual term in the CCA that right may be conditional – *e.g.* it must include an OSI approved licence – or unconditional, depending on the nature of the Project.

CCAs may also extend this right to a Project without assignment and these agreements often include the right to change the licence. Whether assignment or licence, the ability to license under an outbound licence of many different flavours can lead to concerns amongst contributors. When

contributors make a contribution, they tend to know what the licence the software Project they are committing to is on and may be relying that their contribution will also be released under that licence.

Some contributions deal with this by making a commitment to contributors to release the contribution initially under the Project licence current when the contribution is made. Once software is released under a FOSS licence that licence cannot be reneged upon and the software will continue to be released under that licence. This licence commitment does not, however, remove the right for a Project to release that or later versions of the software under different licences, often in parallel with the original licence. This is often referred to as “dual licensing”. It may involve the same software being distributed under more than one FOSS licence or a FOSS and any other including a closed source licence. Many CCAs include statements around this, known as the "outbound commitment".

If a project is started with joint copyright, in England at least, that is very hard to unravel ownership issues in the future.

Examples of Issues Which Harmony Intends to Address

In my experience, discussions starting “IANAL, but...” frequently involve the following issues. We intend to address these through Project Harmony both by producing appropriate wording in the CCA, where appropriate, and addressing the issue in the FAQs.

"There is a minimum amount of code required to attract copyright" This is untrue. The question we need to consider in establishing if copyright applies is whether there is an original work in the UK and in the US, in most cases is a three point test, discussed in depth in the New York law school article.

This is not purely a question of size, but concerns content and originality of the code. Copyright does not consider if the code is inventive and does not protect facts, ideas or systems and methods of operation. In the UK copyright arises once that creative work is fixed in some medium, *i.e.* is somehow written down. So the code in your head is not protected, but once it is typed up copyright automatically arises.

Work for Hire

This is a US concept. In the US, the commissioner of a work owns the work. In the UK, this is not the case (at least not for copyright): the creator owns it unless there is an assignment or the work is created in the course of employment.

Moral Rights

The right to be identified as an author. The UK excludes these by statute with respect to software.

Public Domain

The ability to dedicate works to the public domain exists in the US but this ability does not extend to all places outside the US. For example, in UK, it is generally accepted that copyright works can only enter the public domain once their relevant term has expired. Pending such expiry, copyright cannot disappear and there is always a copyright owner (even if it is impossible to determine who it is).

Need to register Copyright

In the UK at least. Even US lawyers I have discussed this with have found it confusing. In some countries, such as the US there are registration requirements but these don't exist in the UK.

The laws of copyright are not be the same in every country in the world, despite the prevalence of the Berne Convention, and other international treaties. Even though the law of copyright as it applies to computer programs is subject to the Computer Programs Directive 2009/24/EC in Europe, there are still significant differences in copyright law and practice from one European country to the next.

Key Issues in CCAs which are being considered by Project Harmony

The intention here is not to provide a detailed report on our discussions or findings so far, but instead, to set out a brief overview of some of the matters being considered.

- What is contributed, by whom and to whom
individual or company making the contribution?
if an individual, (1) is the contribution as an employee or not and (2) confirmation that the signatory has the right to make the contribution
the rights of minors to make contributions
- IP - Intellectual Property rights in code
- What is being contributed - code or more?
- How is it contributed - only stated contributions or flag if not a contribution
- How is it legally contributed - assignment, exclusive licence, non-exclusive licence
- What if any, commitment is made to the contributor with respect to the distribution of their contribution, (i.e any outbound commitment)
- What if any commitment should be made by contributor re IP eg warranties and indemnities
- Trade Mark – should any trade mark rights be granted or accepted? (We are reaching the conclusion that trade marks are irrelevant in the context of a CCA).
- Copyright law - see above

- Patents - CCAs may rely on an implied patent licence that runs with the code and a personal obligation not to sue us for breach of their patents existing at the time of the contribution or in the future (called a hold harmless or non-assert) from the contributor. At present any need for a patent licence, a hold harmless or non assert or duty to notify of an infringement are all under discussion. For example, should the scope of the non-assert or hold-harmless be limited to patents owned or controlled by the contributor personally, or extend to the entity he or she is employed by?)
- Boiler Plate
- governing law
- excluding the US and UK: implied warranties of merchantability/ satisfactoriness and fitness for purpose
- any need for termination

Project Harmony Process and Participation

The documentation goal is to launch our output to delegates at the European Legal Network's conference on 8 April 2011, and simultaneously to the public through the Project Harmony website. After this time it will be more appropriate to discuss in detail the discussions and reasoning of the Harmony Project in some depth.

Participating in Project Harmony is open to all. We have a network of around 80 participants, being a mix of lawyers and developers. Mark Radcliffe of DLA Piper has taken over the mantle of drafter from SFLC, who led drafting through the first half of this process. Both SFLC and Mark Radcliffe have been in the role of third party editor and cannot dictate or choose the terms. The content of the document is something that the Harmony Project is collectively responsible for..

We hold weekly legal drafting meetings on Wednesdays at noon Eastern Time, which all can participate in, and the draft is available for comment to all participants through coment. Meetings are held by conference call and IRC.¹

Face to face meetings have been helpful to advance drafting and to ensure everyone is aware of the latest developments. Meetings have already been held in Boston, London and Palo Alto, and two more are scheduled for 2nd February and 2nd March in the lead up towards a presentation of outcomes at the European Legal Network Conference in early April.

Project Harmony does not have a CCA.

Our only rule is the Chatham House Rule, means that although Harmony's activities can be freely discussed, no comment may to any individual or organisation. We have a Wiki which stores our meeting minutes, related articles and our thoughts. The Wiki is open to all participants. We have a web site at <http://www.copyrighttharmony.org>.

¹ Internet Relay Chat

Anyone who wants to participate should send me an email, amanda.brock@canonical.com with their request and a short Launchpad name which they can obtain by registering on Launchpad.net. This will give you access to the mailing list, the WIKI and the ability to participate in meetings.

We will include an update on our output in the next edition of IFOSSLR.

About the Author

Amanda Brock studied law at the University of Glasgow, New York University and Queen Mary and Westfield College, University of London and is admitted as a Solicitor in England and Scotland. She is General Counsel of Canonical Group and has worked in-house since 1999.

Amanda has lectured and written widely on commercial, IP and IT law in both business and academic contexts. She was on the Editorial Board of Butterworth's Electronic Business Law, is a founding Editor of IFOSSLR and is on the Advisory Board of the Open Source Centre of Excellence at the Centre of Commercial Law Studies, Queen Mary and Westfield, London.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifoSSLR.org>.

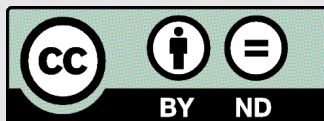
This article should be cited as follows:

Brock, Amanda (2010) 'Project Harmony: Inbound transfer of rights in FOSS projects', *IFOSS l. Rev.*, 2(2), pp 139 – 150
DOI: [10.5033/ifoSSLR.v2i2.49](https://doi.org/10.5033/ifoSSLR.v2i2.49)

Copyright © 2010 Amanda Brock.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Software Patents and Open Source Models in Europe: Does the FOSS community need to worry about current attitudes at the EPO?

*Noam Shemtov*¹

DOI: [10.5033/ifosslr.v2i2.43](https://doi.org/10.5033/ifosslr.v2i2.43)

Abstract

The growing popularity of free and open source software (hereinafter 'FOSS') and the equally significant growth in the number of software patents give rise to the question of whether both models can live in peace, side by side, or does the latter pose a threat to the former. The focal point of this paper is the scope of patent protection to software products in EPC countries, particularly in the light of the FOSS development environment.. Software patents are of great interest to the FOSS community. Modern computer programs are complex. They are developed incrementally where each software generation builds upon the previous one. Avoiding an inefficient re-invention of the wheel, programmers are typically combining thousands of different programs and algorithms in an innovative manner in order to produce the requisite final product. Here is where software patents become relevant. If a large number of those fundamental building blocks are unusable due to patent protection, a type of a 'tragedy of the anticommons' emerges. Hence, developers that may reasonably rely on FOSS licenses to incorporate the licensed software into their programs may later on realize that their actions amount to patent infringement. Obviously, this might have a chilling effect on the development and adoption of FOSS products.

Keywords

Patents; European Patent Convention; Free and Open Source Software; Software Patents;

Info

This item is part of the [Articles](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement. This article has been independently peer-reviewed.

¹ Noam Shemtov is a lecturer in IP and Computer & Communication Law in the Centre for Commercial Law Studies, Queen Mary, University of London.

Introduction

The growing popularity of free and open source software (hereinafter ‘FOSS’) and the equally significant growth in the number of software patents give rise to the question of whether both models can live in peace, side by side, or does the latter pose a threat to the former. The focal point of this paper is the scope of patent protection to software products² in EPC countries³. But before one ventures into the realms of patent law, it may be useful to recall why is it that software patents are of significant interest to the FOSS community. Modern computer programs are complex. They are developed incrementally where each software generation builds upon the previous one. Avoiding an inefficient re-invention of the wheel, programmers are typically combining thousands of different programs and algorithms in an innovative manner in order to produce the requisite final product. Here is where software patents become relevant. If a large number of those fundamental building blocks are unusable due to patent protection, a type of a ‘tragedy of the anticommons’ emerges. In other words, integrating all those programs and algorithms may involve the use of patented products or processes. Hence, developers that may reasonably rely on FOSS licenses to incorporate the licensed software into their programs may later on realize that their actions amount to patent infringement. It is not surprising, therefore, that FOSS proponents are concerned that the possibility of these looming patents may have a chilling effect on the development and adoption of FOSS products.⁴ The question is therefore whether some of the aforementioned building blocks could be locked away under proprietary patent locks and keys.

How it all started: A snapshot of the U.S. position

As far as sheer numbers are concerned, the answer appears to be positive. By 2007, the United States Patent and Trademark Office had already granted 200,000 patents that may be termed as software patents⁵, and it continues to issue such patents at a rate of 20,000 per year.⁶ In the FOSS context, one study found that Linux operating system potentially infringes 283 software patents.⁷ Thus, the software-related legal landscape is already peppered with various patents and appears to get more congested by the day. In order to assess whether software patents pose a threat to the FOSS model it is necessary to examine the nature of such patents and the circumstances under which they may be granted. The answer, of course, is jurisdiction-dependent. It is the latter issue

2 As discussed below, a more appropriate terminology in the European context is ‘computer implemented’ inventions.

3 Patent law is not harmonised across the European Union. Thus, mentions in this paper to patentability in Europe refer to European Patent Organisation (EPO), which is an intergovernmental organisation that was set up on 7 October 1977 on the basis of the European Patent Convention (EPC) signed in Munich in 1973.

4 Greg DeKoenigsberg, *The Red Hat Patent Promise: Encouraging Innovation*, RED HAT MAG., Nov. 2004, <http://www.redhat.com/magazine/001nov04/features/patents/>.

5 i.e. patents granted over software programs designated as ‘products’ or ‘processes’ implemented via software programs, including, for example, the automation of business methods.

6 James Bessen & Robert M. Hunt, *An Empirical Look at Software Patents*, 16 J. ECON. & MGMT. STRATEGY 157, 158 (2007). DOI: [10.1111/j.1530-9134.2007.00136.x](https://doi.org/10.1111/j.1530-9134.2007.00136.x)

7 Press Release: Open Source Risk Management, Results of First-Ever Linux Patent Review Announced, Patent Insurance Offered by Open Source Risk Management, <http://www.osriskmanagement.com/press-releases/press-release-080204.pdf>

that stands at the centre of this paper and in particular software patentability in Europe.

Software patents have proliferated in the last twenty years. The trend started in the United States. It was the Supreme Court of the United States that first grappled with the issue of patentability of software related inventions in a series of three landmark cases.⁸ While it is a long established principle that a pure mathematical algorithm cannot be patented,⁹ it is less clear how far one needs to move away from the realms of the mere abstract in order to render the subject matter of a patent application patentable. Recently, and after much anticipation, the Supreme Court of the United States has struck down the ‘machine-or-transformation’ test, established by the majority of the Court of Appeal of the Federal Circuit in *re Bilski*, as the sole test of software patentability.¹⁰ The Supreme Court returned the ‘machine-or-transformation’ test to its original status as a ‘useful and important clue, an investigative tool’ for establishing eligibility. Hence, it is now clear that software methods as well as business methods are eligible for patent protection in the U.S., as long as the claims are drafted so that the claimed subject matter cannot be said to be purely abstract. Of course, an applicant still has to satisfy the requirements for novelty and non-obviousness.

Thus, at least as far as the U.S. is concerned, software patents are a fact of life that the industry, including the FOSS community, must learn to live with. But how did we get to this stage? Why does the industry wilfully shackle itself in this manner? After all it is not only FOSS friendly entities that bear the consequences.¹¹ In fact, during the Nineties some of the companies that have recently attempted to enforce their software patents against various parties expressed somewhat different views in this context. For example, in 1991 Bill Gates, the founder of Microsoft Corporation, referring to software patents, stated:

*"If people had understood how patents would be granted when most of today's ideas were invented and had taken out patents, the industry would be at a complete standstill today."*¹²

Oracle's statement, submitted to the hearings on software patentability at the US Patent Office in 1994, reads:

"Oracle Corporation opposes the patentability of software. The Company believes that existing copyright law and available trade secret protections, as opposed to patent law, are better suited to protecting computer software developments. Patent law provides to inventors an exclusive right to new technology in return for publication of the technology. This is not appropriate for industries such as software development in which innovations occur rapidly, can be made without a substantial capital investment, and tend to be creative combinations of previously-known techniques."

8 *Gottschalk v. Benson*, 409 U.S. 63, 73 (1972); *Parker v. Flook*, 437 U.S. 584, 596 (1978); *Diamond v. Diehr*, 450 U.S. 175, 193 (1981).

9 For the US, see, for ex', *Le Roy v. Tatham*, 55 U.S. 156, 175 (1852); for the EPC, see Art. 52(2)(a).

10 *Bilski v. Kappos*, No. 08-964 (US June 28, 2010)

11 See, e.g. NTP action against Research in Motion for infringement of its software patents. The case was settled in March 2006, with RIM paying NTP \$612.5 million to make NTP go away, *NTP, Inc v. Research in Motion, Ltd* 418 F.3d 1282 (2005). In 8 July, 2010 NTP filed a lawsuit Apple, Inc., Google Inc., HTC Corp., LG Electronics Inc., Microsoft Corporation, and Motorola, Inc. in the United States District Court for the Eastern District of Virginia for infringing NTP's eight patents related to the delivery of electronic mail over wireless communications systems.

12 L. Lessig 2002-07-24: Keynote to OSCON

Viewed today, these statements are difficult to reconcile with the above companies' patent enforcement policy.

The reasons for this sudden change of heart are multiple. Initially, the great run to patent law started after U.S. courts began to erode the scope of protection available to computer programs under copyright law in the early nineties. Thus, for example, in *Computer Associates v. Altai* the court restricted the scope of protection available to computer programs under the Abstraction-Filtration-Comparison test.¹³ A couple of years later, the court in *Lotus v. Borland* suggested that the available scope of protection should be trimmed even further as the initial inquiry of a court in cases involving copyright infringement of computer programs must be whether the subject matter taken amounts to idea, procedure, process, system or method of operation.¹⁴ As various software companies flocked to the USPTO, other software companies found it more and more difficult to remain passive. Thus, it can be said that many of the parties seeking patent protection nowadays are doing so for reasons that have little in common with the objectives and public policy considerations underlying patent law.¹⁵ To name but a few, nowadays patent protection is often sought for reasons such as attracting investors - by indicating that the company is a significant technological player,¹⁶ as a defense strategy - being able to cross license patents with other market player¹⁷ and participation in a patent pooling model.¹⁸

Whatever the reasons for obtaining such patents might be, as HP's Vice President of Linux, Martin Fink, has said: "[a]t the end of the day, software patents are a way of life. To ignore them is a little bit naive."¹⁹ Whether we like it or not, the software community must learn to deal with the availability software patents in the U.S.

But are software patents 'a way of life' in Europe too? Can one obtain a 'software patent' in Europe in a manner similar to that available in the U.S.? It is a common wisdom that the situation in Europe is not similar to that in the U.S., in that it is more difficult to obtain patent protection over software products in Europe. But what exactly does that mean? Is it possible for one to obtain a patent for a software product in Europe and if so, under which circumstances?

EPC: Legislative Framework

Prior to my examination of European jurisprudence regarding patent protection for software products, it should be noted that patent law is not harmonized at European Union level.²⁰ Thus,

13 *Computer Assocs. Int'l v. Altai*, 982 F.2d 693, 705 (2d Cir. 1992). The court found that the part taken by the defendant did not amount to the plaintiff's program's core protectable expression.

14 *Lotus Dev. Corp. v. Borland Int'l*, 49 F.3d 807, 819 (1st Cir. 1995). Here the court refused to hold that a program's menu command hierarchy was a protectable subject matter under copyright law.

15 Large software companies are not likely to gain meaningful financial rewards from enforcing their patents; see, e.g., Ronald J. Mann, *Do Patents Facilitate Financing in the Software Industry* 83 TEX. L. REV. 961, (2005), at 996.

16 *Ibid.*, 993-995.

17 *Ibid.*, 990

18 Patent pooling is a derivative of the cross licensing concept. See, e.g., Open Invention Network, <http://www.openinventionnetwork.com/about.php>

19 Stephen Shankland, HP: Don't Like Software Patents? Learn to Deal, ZDNET NEWS, Feb. 15, 2005, <http://news.zdnet.com/2100-3513-22-141325.html>.

20 There are a few exceptions in specific areas where harmonising measures were taken on an EU level; see, e.g., The Biotechnology Directive (98/44/EC).

references in this paper to European patent law or European jurisprudence are not made in a European Union context. Although the European Commission sought in the past to harmonize patent law within European Union with respect to software patents, this attempt was overwhelmingly rejected by the European Parliament in 6th July, 2005.²¹ Although EU harmonization in this context may return in the future to the public agenda under initiatives such as Community Patent or the European Patent Litigation Agreement (EPLA), this paper addresses the question of patentability of software products under the present legislative framework; i.e. the European Patent Convention (EPC). The EPC is an intergovernmental treaty that is independent of the European Union; its members extend beyond membership of the EU.²² The EPC is mainly concerned with granting European Patents.²³ This was facilitated by the establishment of the European Patent Office (EPO). When the EPC was formulated it was decided that in order to have an effective granting process it was also necessary to harmonise the basic national rules of patent law; i.e. rules regarding patentability and validity. Thus, rules concerning patentability and validity are harmonised both at EPO and national law levels. On the other hand, rules concerning issues such as infringement, enforcement, renewal, revocation and litigation are governed by national law. It is noteworthy that the EPC being independent of the EU also means that decisions of EPO Boards of Appeal are not, strictly speaking, binding on national courts but could be described as being of highly persuasive authority.²⁴

Software patents and computer programs ‘as such’

‘Software patents’ is a term best avoided in the context of patentability of computer programs in Europe. For reasons discussed below it is suggested that ‘computer implemented inventions’ is a more appropriate term. In order to obtain a patent, an applicant is required to establish that: (a) the claimed subject matter relates to an invention,²⁵ (b) the said invention is novel,²⁶ (c) it involves an inventive step²⁷ and, (d) has an industrial application.²⁸ Article 52 deals with patentable inventions. Inter alia, it provides a list of ‘non-inventions’, i.e. a list of subject matter that are to be considered as falling outside its definition of ‘invention’ and hence non-patentable. Thus, an application claiming a subject matter that falls under the scope of the non-invention list is not patentable and no enquiry as to its novelty, inventive step or industrial application needs to be carried out. Article 52(2) states, inter alia,²⁹ that “mathematical methods ..., schemes, rules and methods for performing mental acts, playing games or doing business, programs for computers and the presentation of information” are not to be considered as inventions. However, Article 52(3)

21 On 6th June 2005 the European Parliament rejected Commission proposal [COM\(2000\)0199](#) (a proposal for a [European Union directive](#) aimed to harmonise national [patent](#) laws and practices concerning the granting of [patents for computer-implemented inventions](#)) by 648 votes to 14 with 18 abstentions.

22 As of 1 May, 2010 the EPC consisted of the following 37 member states: Albania, Austria, Belgium, Bulgaria, Switzerland, Cyprus, Czech Republic, Germany, Denmark, Estonia, Spain, Finland, France, United Kingdom, Greece, Croatia, Hungary, Ireland, Iceland, Italy, Liechtenstein, Lithuania, Luxembourg, Latvia, Monaco, FYROM, Malta, Netherlands, Norway, Poland, Portugal, Romania, Sweden, Slovenia, Slovakia, San Marino, Turkey.

23 Giving rise to a portfolio of national patents that are in force in EPC member states designated by the applicant.

24 Thus, a national court deciding on a domestic patent application is not bound by the approach taken at the EPO, although it is nevertheless more likely to follow it than not.

25 EPC, Art. 52.

26 EPC, Art. 54.

27 EPC, Art. 56.

28 EPC, Art. 57.

29 The ‘non inventions’ list under Art. 52 also covers scientific theories, discoveries and aesthetic creations.

provided that the list of excluded subject matter provided for in Article 52(2) is so excluded only to the extent the excluded thing is claimed ‘*as such*’.

It is suggested that it is the aforementioned text of the EPC that gives rise to the ongoing uncertainty of the scope of the exclusion from patentability of computer programs. The meaning of ‘*as such*’ in the context of computer programs has been anyone’s guess during the past two decades. The ensuing uncertainty has been illustrated by a series of decisions of different EPO Boards of Appeal, a referral to the Enlarged Board of Appeal by the President of the EPO, as well as by a number UK court decisions seeking to apply the UK equivalent of Article 52. Not only were those English decisions irreconcilable with those of the EPO, but to a certain extent the said EPO decisions appeared to be irreconcilable with one another.

Computer-implemented inventions

It is clear from the text of Article 52 that computer programs ‘*as such*’ cannot be claimed in a patent application. Thus, a claim to a computer program in the narrow sense of the word, e.g. which literally claims computer executable instructions or an executable software module clearly falls under the exception of Article 52(3). However, things become less clear where claims are not literally formulated as aforementioned, but are rather directed to computer systems, computer-implemented methods, computer program products and the storing of computer programs. It should be noted that the underlying method of performing the latter type of claims is often identical to that of performing the former one. Rather than using ‘computer programs’ or ‘software’, for the purpose of clarity ‘computer-implemented’ inventions is used as a term to encompass inventions the implementation of which involves the use of a computer, computer network, or other programmable apparatus, with features realized wholly or partially by means of a computer program. Thus, the main question is when does an application for a computer-implemented invention relate to a computer program “*as such*”, thus becoming a ‘non-invention’ under Article 52³⁰?

The starting point of the present discussion is the Technical Board of Appeal landmark decision in *Vicom*.³¹ The case concerned claims to a method of digitally processing images and to an apparatus, which may be a general-purpose computer, for carrying out that method. The Technical Board of Appeal stipulated that an invention is patentable if it satisfies the normal requirements for patentability under the EPC and should not be prejudiced against simply because its implementation required modern technical means in the form of a computer program. The Board then stated: “decisive is what *technical contribution* (emphasis added) the invention as defined in the claim when considered as a whole makes to the known art.” This later became known as the ‘technical contribution approach’. According to this approach, when examining a patent application in this context, one should disregard the form or kind of claim and concentrate on its content in order to identify the real contribution which the subject-matter claimed, considered as a whole, adds to the known art. If this contribution is not of a technical character (i.e. if it falls

30 However, this is not the only relevant question; the claimed invention’s ‘technical character’ is of significant importance as well.

31 *Vicom* (1986) T208/84, [1987] OJ14.

exclusively within one of the excluded areas³²), there is no invention within the meaning of Article 52(1).

The first step of departure from the so-called technical contribution approach was in T1173/97. In this case the examining division found that there was an 'invention' and was prepared to grant a patent in respect of claims that were accepted under Vicom's technical contribution approach, since the application manifested technical contribution. However, the application at stake also included claims directed not to the system as a whole or a method of operating the system as a whole, but to a program, in two forms, as follows:

"A computer program product directly loadable into the internal memory of a digital computer, comprising software code portions for performing the steps of [independent method] claim 1 when said product is run on a computer," and "A computer program product stored on a computer usable medium, comprising: computer readable program means for causing a computer to [carry out the various steps of method claim 1]." The technical board of appeal held that a computer program, claimed on its own, is not excluded from patentability if the program, when running on a computer or when loaded into a computer, brings about a technical effect which goes beyond usual physical interaction between the program and the computer."

Rather than looking for a technical contribution to the state of the art as in Vicom, the board stated that computer programs must be considered as patentable inventions where they have a technical character. This 'technical character' or 'technical effect' should be understood as a further technical effect in that it goes beyond the normal technical effect that is inherent to a software-hardware environment. The board emphasized that in this context it made no difference whether or not the program was claimed by itself or as part of a system. Thus, it was not the wording of the claims that matter but rather their content. According to the board, when examining the content of an application for a 'technical effect', the inquiry should be conducted independently of the prior art. Thus, the so called further technical effect does not have to be novel.

The abandonment of the 'contribution approach' was further elucidated in the subsequent Pension Benefit Systems decision.³³ In that case, the objection raised by the European Patent Office was that the claims in the application related to a computerized method of doing business, a category excluded from patentability under Article 52 of the EPC. The examination division maintained that, since the contribution that the claimed subject-matter made to the known art was solely within the 'business' field, the contribution is to be regarded as non-technical and the application should be refused on the basis that the claimed subject matter did not constitute an invention. In the Technical Board of Appeal's view, the 'contributions approach' confused the requirement of 'invention' with the requirements of 'novelty' and 'inventive step', since it looked at the contribution made by the invention to the prior art. According to the board, an inquiry in relation to the state of the art has no place in an examination under Article 52 and should be conducted only with relation to an examination for novelty and inventive step. Thus, the board held that an apparatus (i.e. a computer loaded with the program) for carrying out an activity excluded as such from patentability by Articles 52(2) and (3) EPC was not, in itself, excluded from patentability

³² An excluded subject matter is regarded, by definition, as non-technical.

³³ T931/95 (OJ EPO 2001, 441)

under Article 52.³⁴

However, that decision did not extend the same logic to methods employing technical means. While the apparatus claims were held to comply with the requirements of Article 52, the corresponding method claims were held not to employ technical means and thus fall foul of Article 52. It should be noted that the apparatus claims in *Pension Benefits* did not get far. The board in *Pension Benefits* held that ultimately the application was bound to fail under Article 56 for lacking inventive step since the contribution it made to the known art (i.e. to the field of ‘doing business’) lay solely within an excluded field. Thus, the board held that where the contribution is made exclusively within an excluded field, then by definition it does not possess a technical character. Where a feature of a claim does not have a technical character or a technical effect, it is deemed to be within the knowledge of the skilled person for the purpose of assessment under Articles 54 and 56.

The decision of the board in *Pension Benefits* on the non-compliance of the methods claims with the requirements of Article 52 was explicitly rejected by the decision of the board in *Hitachi*³⁵. Thus, in *Hitachi* the technical board of appeal followed the board’s decision in *Pension Benefits* on the apparatus claims but rejected it on the method claims. The board held that claims involving technical means were not excluded from patentability under Article 52. It followed that claims directed to a method of operating a computer involved a computer, which is a physical object of technical nature, and thus could not be excluded from patentability as non-inventions.

The decision in *Microsoft*³⁶ extended this logic even further. In this case the board held that claims to a program (‘computer executable instructions’) on a computer readable medium also avoided Article 52 exclusion. The board reasoned that the subject matter of the contested claim had a technical character since it related to a computer readable medium, which is a technical product involving a carrier. Thus, while T1173/97 suggested that it was not the wording of the claims that matter but rather their content, *Microsoft* clearly stands for the proposition that in order to avoid Article 52 exclusion, an applicant merely needs to make sure that claims to computer programs³⁷ should be drafted so as to explicitly mention the use of computer or computer readable storage medium. Since a computer or a carrier is a ‘technical product’, the application would manifest the requisite technical effect in order to avoid Article 52 exclusion. Similarly to the approach taken in *Pension Benefit* and followed in *Hitachi*, the technical Board in *Microsoft* shifted the focus of its examination to the assessment for inventive step. Thus, while any technical means was sufficient to overcome the exclusion of Article 52, a ‘further’ technical means was still required for the assessment for inventive step. It is features of the claims that contribute to the ‘further’ technical character of the claimed subject matter that are taken into account for the assessment of inventive step. Since on the facts of the case the Technical board found that such further technical effect was present and non-obvious, the appeal was allowed.

The approach taken by the technical board in *Microsoft* has not been challenged by any

34 Since the board held that the apparatus in question was not a computer program nor a method of doing business ‘as such’, it was an ‘invention’ within the meaning of Article 52(1).

35 T258/03.

36 T424/03.

37 As mentioned, claims to computer programs in the narrow sense of the word are excluded from patentability under Article 52.

subsequent decision and may be considered as representing the present legal position in the EPO.³⁸ Subsequent decisions further elucidated and developed this approach. In *Duns*³⁹ the technical board of appeal concluded that some features of a claim which, when considered alone, might fall under the exclusion of Article 52, could nevertheless be taken into account for the assessment of inventive step.

A shift in focus of assessment of patentability of ‘computer implemented inventions’: Art. 56

As we have seen, computer implemented inventions find it relatively straightforward to pass the patentability hurdle of Article 52 under the currently prevalent approach of the EPO boards of appeal. Indeed, according to this approach any technical means will do in order to render a claimed subject matter ‘an invention’ and thus comply with the requirements of Article 52;⁴⁰ it is not surprising therefore that many refer to the said approach as the “any hardware” or “any technical means” approach.⁴¹

This, however, does not mean that such inventions are easily patentable and that the area in which the contribution made to the state of the art is of no relevance. This factor is still of crucial importance to the invention’s patentability. But under the ‘any hardware’ approach, the said factor is now evaluated under the assessment for inventive step. As mentioned, features that lay exclusively in an excluded field are not considered as having a technical effect, while features that are held to be not within an excluded field may have a technical effect and could therefore be taken into consideration when assessing inventive step. However, it should be noted that the former type of features may still have a role to play under an inventive step assessment.

This was made clear in *Duns*,⁴² where the technical board of appeal was required to assess the patentability of an application, the main request of which concerned a method for estimating sales activity of a product at sales outlets. The board explained that when examining patentability of an invention in respect of a claim, the claim must be construed to determine the technical features of the invention, i.e. the features which contribute to the technical character of the invention. Although it is legitimate to have a mix of technical and non-technical features appearing in a claim, the board noted that novelty and inventive step can be based only on technical features, which thus have to be clearly defined in the claim. The board stated that non-technical features, to the extent that they do not interact with the technical subject matter of the claim for solving a technical problem,⁴³ i.e. non-technical features “as such”, do not provide a technical contribution to the prior art and are thus to be ignored in assessing novelty and inventive step. However, according to the board, such non-technical features may nevertheless be used for the purpose of the problem-and-solution approach that was developed as a test for whether an invention meets the

38 See G3/08, 10.8.2

39 T154/04 (OJ EPO 2008, 46)

40 This is so since Article 52(2) EPC does not exclude from patentability any subject matter or activity having technical character, even if it is related to the items listed in this provision since these items are only excluded “as such”.

41 See, G3/08, 10.6

42 *Supra*, 39.

43 The general approach in the EPO for assessing inventive step being the ‘problem and solution’ approach.

requirement of inventive step, where the problem must be a technical problem.

Defining the technical problem without referring to non-technical features, however, may be difficult where the actual novel and inventive concept making up the core of the claimed invention resides outside any technological field, as it is frequently the case with computer-implemented inventions. Thus, defining the problem without referring to this non-technical part of the invention, if at all possible, will generally result either in an unintelligible vestigial definition, or in a contrived statement that does not adequately reflect the real technical contribution provided to the prior art.⁴⁴ The Board therefore allowed the “aim to be achieved in a non-technical field to appear in the formulation of the problem as part of the framework of the technical problem that is to be solved.”⁴⁵ Nevertheless, it is clear that without an inventive technical contribution to the state of the art, technical being in the realms outside the excluded fields as defined under Article 52, an invention may not be patentable. Thus, the crux of the matter in such applications is whether the invention in question manifests a non-obvious technical contribution to the state of the art. A contribution that falls squarely within an excluded field is not ‘technical’ and its obviousness is therefore of no relevance. This is clearly different from the position under U.S. law, where the ‘non-obviousness’ of an invention may reside in what may be termed as non-technical under EPO jurisprudence but may nevertheless render the invention in question patentable under U.S. law.

It follows that in the heart of the question of patentability of computer implemented invention in the EPO lies the concept of technicality, whether referred to as ‘technical effect’ or ‘technical character’. But what does one mean when referring to ‘technical’ in this context? It is clear that the usual interaction between software and hardware is of technical nature. This, indeed, is taken into account when rendering the claimed subject matter ‘an invention’ under Article 52. It will also be taken into account when carrying out an assessment under Article 56. However, if no further technical feature could be identified, such technical features would quite clearly be obvious to the skilled person as they form part of the common general knowledge in this field.⁴⁶ Thus, a technical means that goes over and above the abovementioned usual interaction between hardware and software is needed in order to render an invention patentable. This raises the question of which types of claim features may possess such technical nature. Unfortunately, to date the EPO boards of appeal always refrained from providing a definition to the term technical, even in the form of general guidelines. The same position appears to apply to the assessment of the activity of programming. While it is clear that every instance of programming involves technical considerations since it is concerned with defining a method that could be carried out by a machine, this in itself is not enough to establish that the program that results from the said programming has a technical character. Again, something additional is needed; it should be demonstrated that the programmer had technical considerations beyond merely finding a programming algorithm to carry out the requisite procedure.

44 Supra 39, para 16.

45 Ibid.

46 The skilled person is skilled in the field of information technology, aware of common general knowledge in information technology and having no knowledge of non-technical fields.

In the meantime in England...

English Courts have been grappling with early inconsistent EPO case law for most of the past decade. In this context one must bear in mind that decisions of the EPO boards of appeal are not binding on national courts of EPC countries but are merely persuasive. The first English case that reviewed the EPO trilogy of decisions discussed above, highlighting the difficulties in reconciling them with each other as well as with *Vicom*, was *Aerotel*.⁴⁷ In this case, the Court of Appeal examined the said EPO decisions, identifying a clear inconsistency between their different approaches. Describing the *Pension Benefit* and *Hitachi* approaches as ‘The Lord Giveth, the Lord Taketh away’, the Court of Appeal explained: ‘the giving is the passing of Art. 52(2), the taking away being the device of treating the excluded matter as known’.⁴⁸ In contrast, the *Microsoft/Data Transfer* approach was described as ‘the Lord Giveth but the Lord Doth not Always Taketh away’.⁴⁹ The Court also emphasized that none of the trilogy decisions was reconcilable with *Vicom* and its technical contribution approach. The Court acknowledged the need to maintain a harmonized position within the EPC countries and vis - à - vis the EPO Boards of Appeal. However, it felt that that as there was uncertainty as to which approach prevailed within the EPO itself, it should follow the long standing approach of the English courts—the technical contribution approach, as established in *Vicom* and endorsed by the Court of Appeal in *Merrill Lynch*⁵⁰ and *Re Gale*.⁵¹ The court emphasized its willingness to reconsider its position only in the event of the Enlarged Board of Appeal ruling on the issue and laying the uncertainty within the EPO to rest. It is yet to be seen whether the recent refusal of the Enlarged Board of Appeal to consider the questions referred to it by the then President of the EPO due to lack of inconsistency in the case-law of the boards of appeal would satisfy the English Court of Appeal that the EPO boards of appeal now speak with one voice on the matter.⁵²

Unsurprisingly, subsequent English High Court decisions appeared to follow the same footpath.⁵³ The technical contribution approach got further support when the Court of Appeal revisited the aforementioned issues in *Symbian*.⁵⁴ Similarly to its decision in *Aerotel*, the court reviewed previous English authorities as well as decisions of the EPO boards of appeal and unsurprisingly concluded that English courts are still bound to adhere to the technical contribution approach for assessment under Article 52, while it appeared that the EPO had abandoned that approach and now allows virtually any application involving a computer program to pass the test of Article 52, simply due to the technicality inherent in the interaction between a program and a computer onto which it is loaded. However, using a more conciliatory tone than in *Aerotel*, the court opined that the important thing was that both approaches sought to identify a technical contribution.⁵⁵ Whether

47 *Aerotel Ltd (a company incorporated under the laws of Israel) v Telco Holdings Ltd, Telco Global Distribution Ltd, Telco Global Ltd* [2006]

EWCA Civ 1371

48 *Ibid*, para 28.

49 *Ibid*.

50 *Merrill Lynch's application* [1989] RPC 561.

51 *Gale's application* [1991] RPC 191.

52 G 3/08.

53 E.g., see *Halliburton v Smith* [2005] EWHC 1623 (Pat); *Shoppalotto.com's application* [2005] EWHC 2416 (Pat); *Cappellini's Application* [2007] FSR 26; *Raytheon's Application* [2008] RPC 3; *Autonomy Corporation Ltd* [2008] EWHC 146 (pat)

54 [2008] EWCA Civ 1066

55 Whatever the actual terminology used may be.

one seeks to identify such technical features while making an assessment under Article 52 or one under Article 56 would matter little in most cases as far as the final outcome of the case is concerned.

It is true that whether one examines an application for technical contribution under Article 52 or Article 56 may lead to the same outcome as an application that may be refused by an English court for being a non-invention due to lack of technical features that go beyond these which are inherent in a computer-software environment should, in principle, be refused by an EPO board of appeal for lack of inventive step under Article 56. This conclusion, however, may be valid if one assumes that what may constitute a technical feature before an EPO board of appeal for the purpose of Article 56 is also likely to amount to technical contribution before an English court when examining a case under Article 52. It is arguable whether that is indeed the case.

The amorphous nature of ‘technical character’

The elusive ‘further’ or ‘additional’ requisite technical effect appears to be assessed on a case by case basis rather than by reference to an established set of principles and guidelines. We know that there is no general definition to what constitutes ‘technical’ and we also know that there are clearly grey areas where it would be difficult to anticipate whether such technical effect is present. Can any lessons nevertheless be learned from the case law of the EPO boards of appeal regarding ‘technical character’?

A review of the many instances in which the boards of appeal found computer implemented inventions to have or not to have a technical character and thus manifest an inventive step is beyond the scope of this paper and in any event would be a little benefit as such decisions are limited to their particular facts. However, a number of helpful observations can be made in the context of computer implemented inventions. It appears that the term ‘technical’ may include features such as the processing of physical data parameters or control values of an industrial process, physical features of an entity (e.g. memory ports) and perhaps most relevant for the FOSS community, processing which *affects the way a computer operates* (e.g. saving memory, increased speed, security of a process, the rate of data transfer etc’). There are, however, considerable grey areas and the outcome of many applications for computer implemented inventions is far from certain.

A favourite example of the author of this paper to the uncertainty concerning the outcome of applications for computer implemented inventions in this context is a fairly recent but scarcely cited decision of the technical board of appeal in *Circuit simulation/Infineon Technologies*.⁵⁶ The application at stake concerned a computer implemented simulation or modelling method for testing the performance of an integrated circuit under the influence of a 1/f noise. It was based on a mathematical formula that generated random numbers that produced an exact 1/f noise into the simulation. The claimed method required shorter computing time and less storage space in designing integrated circuits. The claimed method accordingly made it possible to simulate noisy circuits on smaller computer systems, which were previously not powerful enough for that purpose, and to simulate large circuits, which previously could not have been simulated on any

⁵⁶ T1227/05.

computer system whatsoever. In addition to claiming the method, the application contained claims for a computer program executing the method, a data medium holding the program, and a computer system on which the program was loaded. Unsurprisingly the board of appeal overturned the examining division decision to refuse the application on the ground that the simulation method claimed in it constituted a mental act or a mathematical method as such and was thus excluded from patentability under Article 52(2) as a non-invention. More interestingly though were the board's findings regarding the assessment for inventive step. It is noteworthy that the claimed method did not improve or affected the internal operation of a computer per se. As mentioned, at the heart of the application was an algorithm that made the simulation of an integrated circuit's design more effective. Thus, in essence, the claimed method consisted of a mathematical formula implemented in a computer program for simulating noise-affected circuits. The only thing that the claimed method produced was data, which was employed at a later stage, separately from the claimed method, in the physical activity of integrated circuit fabrication. Hence, the relevant industrial process, i.e. the fabrication of integrated circuits, was not part of the claimed method and was carried out at a different time and quite often at a different location from the simulation stage. Nevertheless, since according to the board of appeal the claimed method was a practical and practice-oriented part of the electrical engineer toolkit, rather than being a purely mathematical theory or a mental act, all steps relevant to the circuit simulation, including the mathematically expressed claim features, contributed to the technical character of the claimed subject matter. Thus, having the requisite technical effect, the said features were taken into consideration when assessing the invention for inventive step and resulted in the board's finding that the invention at stake manifested inventive step.

A careful examination of the board's decision reveals what might have been one of the main reasons for the board's finding: a public policy based analysis of the market for integrated circuits! Thus, analysing the present state of technology and needs of the twenty first century engineer, the board noted that an increasing number of fields in the engineering science use numerical simulation methods as a cost-effective alternative to expensive experimental investigations, which consume significant time and personal resources. In many industrial fields numerical simulation methods evolved to be a key technology and should, therefore, be eligible for protection. Accordingly, in the case before it, the Board held that the claimed simulation method must be regarded as a modern technical method, which forms an essential part of the fabrication process and precedes actual production as an intermediary step. In the light of the manner in which the industry is developing, the final implementation of the simulation outcome and the actual manufacturing process involves very little innovative effort if any at all. Furthermore, the Board pointed out that nowadays the development and production stages of integrated circuits are increasingly separated, materially and geographically, in a globally distributed industry. For these reasons, a simulation method should not be denied patent protection merely because it does not involve a physical stage.

It is suggested that this decision is a vivid example of the amorphous nature of the concept of technical character and the uncertainty which surrounds it. In this instance it enabled the board to engage in judicial activism where it explicitly identified what it deemed to be a need of the relevant industry and tailored its definition of what constitutes a technical character accordingly.

Summation

While it is true that software patents are more prolific in the US, it is clearly wrong to assume that no such patents exist or could be granted in Europe due to the effect of Article 52(2) EPC. Although referred to as ‘computer implemented’ inventions rather than software patents, it has been demonstrated that in certain circumstances software features, which may cover aspects of FOSS development projects, are eligible for patent protection in Europe. The exact scope of such instances ultimately hinges on the specificities of the software in a given FOSS project and the meaning the court or board chooses to inject into the term ‘technical’ in each case. What is certain though is that it is not only the US patent regime that the FOSS community should be concerned with; when looking over one’s shoulder, it may also prove wise to check what is brewing at the corridors of the EPO in Munich.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Shemtov, Noam (2010) 'Software Patents and Open Source Models in Europe: Does the FOSS community need to worry about current attitudes at the EPO?', *IFOSS L. Rev.*, 2(2), pp 151 – 164
DOI: [10.5033/ifosslr.v2i2.43](https://doi.org/10.5033/ifosslr.v2i2.43)

Copyright © 2010 Noam Shemtov

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Software Interactions and the GNU General Public License

Malcolm Bain^a

(a) Partner, id law partnes, Barcelona.

DOI: [10.5033/ifosslr.v2i2.44](https://doi.org/10.5033/ifosslr.v2i2.44)

Abstract

This article presents the current results of the work carried out within a working group of the European Legal Network of lawyers, facilitated by the Free Software Foundation Europe, whose aim is to provide some general guidance to lawyers and developers working with free software to understand the technical and (potentially) legal effects of the interaction or interoperation of two programs in the context of GPLv2 licensing.¹

Keywords

GPL; linking; copyright, Free and Open Source Software

Info

This item is part of the [Articles](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement. This article has been independently peer-reviewed.

The so-called “GPL linking” debate has been raging for the last 18 years, and probably will go on for a quite a few more. It has been seriously considered by legal authors such as, among others, Lawrence Rosen in “*Open Source Licensing: Software Freedom and Intellectual Property Law*”²

-
- 1 This article is based on the work carried out in the context of the Software Interactions working group of the Free Software Foundation Europe, of which I am “rapporteur”, and takes from the resulting *Working Paper on the legal implications of certain forms of Software Interactions (a.k.a linking)*”, which is available online at <https://wiki.fsfe.org/EuropeanLegalNetwork/LinkingDocument>>. I would like to thank those participating in this work group for their input and feedback, however all opinions and errors made herein are my own. Special thanks go to Neil Brown, Andrew Katz and Martin von Willebrand for their comments on this paper.
 - 2 Rosen, Lawrence (2004), ‘*Open Source Licensing, Software Freedom and Intellectual Property Law*’, Prentice Hall, available online at <http://www.rosenlaw.com/oslbook.htm>>

and “*The unreasonable fear of infection*”,³ or Dan Ravicher, now of the Software Freedom Law Center, in an LWT interview in 2003, “*Dan Ravicher on derived works*”.⁴ It has been hotly argued on discussion lists such as Debian-legal and in comments on *LWT*, *GrokLaw* or *Slashdot*,⁵ and (more politely?) on Open Source Initiative’s license discussion list⁶ and the FSF-Europe European Legal Network’s own discussion list. In Europe, authors on the subject include Andrew Katz in an article for the Society for Computers and Law, “*GPL – the Linking Debate*”,⁷ and Mikko Välimäki in ‘*GNU General Public License and the Distribution of Derivative Works*’.⁸

The question at the heart of the matter is under what circumstances, if a software program or application “uses” GPL’d code (and I use the vague word “use” on purpose here, as I comment on this below), does this use cause the application to be covered by the copyleft provisions of the GPL – either as a derivative work or otherwise – and thus render any redistribution of the application subject to the GPL.

Art.2b of the GPLv2 provides:

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

...

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

...

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program,

3 Rosen, Lawrence (2001) ‘The unreasonable fear of infection’, *Linux Journal*, available online at <http://rosenlaw.com/html/GPL.PDF>

4 Available online at <http://lwn.net/Articles/62202/>. Other US based articles include Determann, Lothar (2006): ‘*Dangerous Liasons--Software Combinations as Derivative Works? Distribution, Installation, and Execution of Linked Programs Under Copyright Law, Commercial Licenses, and the GPL*’ *Berkeley Technology Law Journal*, Volume 21, issue 4; online at http://www.btlj.org/data/articles/21_04_03.pdf

5 E.g.: on LWN: ‘*GPL and linking*’ (Feb 16, 2006), at <http://lwn.net/Articles/172226/>; Slashdot: ‘*WordPress Creator GPL Says WP Template Must Be GPL’d*’ (July 22, 2010), at <http://yro.slashdot.org/story/10/07/22/1935248/WordPress-Creator-GPL-Says-WP-Template-Must-Be-GPLd>

6 Archives available at <http://www.mail-archive.com/license-discuss@opensource.org/>

7 Katz, Andrew (2007) ‘*GPL - The Linking Debate*’, *SCL Magazine*, Vol 18 Issue 3; available online at <http://www.scl.org/site.aspx?i=c10>

8 Välimäki, Mikko, ‘*GNU General Public License and the Distribution of Derivative Works*’, 2005 (1) *The Journal of Information, Law and Technology (JILT)*, available online at http://www2.warwick.ac.uk/fac/soc/law2/elj/jilt/2005_1/välimäki/.

the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

Understanding this is a key question for any developer who wishes to “use” a GPL component in her own application, as this has an impact not just on the licensing of the resulting work, but also implications for license compliance such as providing source code and a copy of the license, and the resulting negative consequences (legal and/or reputational) for getting it wrong.

This article does not comment substantively on this debate, but mainly reports on the work carried out by the “Software Interactions Working Group” of the aforementioned Freedom Task Force to bring some light to the matter. The substantive work, which we will call for ease the “Software Interactions Document”, focuses on the interpretation of GPLv2 and has been presented and published as a work-in-progress by the FSFE.⁹

1. Presenting the Software Interactions Document

1.1 Purpose and scope

The aim of the Software Interactions Document is to provide some general guidance to lawyers and developers working with free software to understand the technical and (potentially) legal effects of the interaction or interoperation of two programs. While there is a general awareness of the issue among serious users of free software, we have found there are a lot of misconceptions, both in the legal and IT engineering communities, regarding the scope, impact, effect and obligations surrounding the use of GPL'd software. The document aims to clear up some of these misconceptions and note consensus if and where there is consensus on any aspect, and highlight areas of debate that may usually be linked to the specifics of each case.

More specifically, the purpose of the work is to facilitate understanding of different mechanisms of interaction between programs in order to assist decision making, in free and non-free software projects, for intermediaries within the software supply chain and end-users, who use or intend to use GPL'd software programs, as to whether a program may or must be considered a derivative work of another (original) work, or possibly a collective (composite) work incorporating a

⁹ *Working Paper on the legal implication of certain forms of Software Interactions (a.k.a linking)*, available online at <<https://wiki.fsfe.org/EuropeanLegalNetwork/LinkingDocument>>. The analysis is mainly based on the European legal framework established by Council Directive of 14 May on the legal protection of computer programs (91/250/EEC) Official Journal L 122 , 17/05/1991 p. 42-46 (“EUCPD”, consolidated in Directive 2009/24/EC Official Journal L 111 , 5/5/2009 p. 16-22).

previous work, or whether it could be considered independent. Even more specifically, it aims to shed some light on the use of GPLv2'd software components, or creating software for GPLv2 platforms, and the scope of the copyleft provisions as established in this license.

The document is descriptive and exploratory, focussing on a limited number of interaction mechanisms,¹⁰ and it does not aim to establish any legal or normative position or “doctrine” in the matter – it presents a step by step legal analysis of the combination of two software components and the considerations which could or would be taken into account, as we describe below. The actual legal effect of any form of interaction will depend on the circumstances of each case, and the work only provides preliminary (and simplified) examples of code.

In addition, the legal interpretation and consequences of any form of interaction (e.g. whether it creates a derivative work or not, under which license a program may be distributed, what are the distribution obligations) will depend on the specific legal framework of the jurisdiction (state) in which the question arises, whether during the course of developing new software or in copyright infringement proceedings. For example, certain jurisdictions may not grant copyright protection for certain aspects or elements of a work (e.g. in the USA, “*processes, systems and methods of operation*”,¹¹ in the EU member states, “*ideas and principles... underlying [a program's] interfaces*”¹²) which may limit the scope of exclusive control of a copyright holder.

Finally, the document also includes comment on the so-called “community view” – i.e. the opinion of the members of the community from which the software is taken or in which the software is developed, taking into account that there is not necessarily a single representative community voice. Where possible, we have tried to indicate where there is a divergence. Due to the nature of the free software environment, from a business point of view the community view may be of equal if not more relevance than the strict legal interpretation of a license, for the purpose of assessing risks and benefits when taking a decision about the licensing and distribution of inter-related components of software.

1.2 Challenges

Work on the document has been no easy task, with a number of challenges.

First, we have found that there is no clear technical definition or consensus on certain (or any!) forms of software interactions, with many forms of implementation, exceptions, special cases, contexts, programming paradigms and languages.

Second, each of the persons participating in the work has brought a different legal background and tradition to the table, with a different approach to asking and answering questions towards resolving the issue, and a different legal vocabulary and set of case law. While copyright law has

10 The document currently looks at static and dynamic linking, Remote Procedure Calls, system calls, macro and template expansions, “plug-ins” and interpreted language communication mechanisms. As technology evolves and further input is provided, the aim is to expand the analysis to further interactions, if necessary.

11 Section 102 of the U.S. Copyright Act (title 17 of the U.S. Code). Relevant comment can be found in Samuelson, Pamela (2007) ‘*Why Copyright Law Excludes Systems and Processes from the Scope of its Protection*’, Texas Law Review, Vol. 85, No. 1, available online at <http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1002666>.

12 Article 1.1, Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs.

been “harmonised” to a certain extent internationally,¹³ it is not sufficiently so either to provide a single method of legal reasoning or juridical approach (for example, the US 9th Circuit Abstraction-Filtration-Comparison test¹⁴ would not necessarily - or at all - be used by courts in Spain or France to determine copying or creation of derivative works) nor, even when using the same approach, to come to a single interpretation of the law to a theoretical series of facts.

Third: the “Community” view. Decisions relating to the use or non-use of GPL’d code, like most decisions, are not based purely on legal arguments, but also significantly on a risk analysis that takes into account the views of the free software community as a whole (which may eventually be consecrated as a “trade custom”) and by the community of the specific GPL’d software that is to be used. There is a difference between dealing with a vociferous, if not necessarily legally correct, community (and possibly just a minority of members) and one where there is space to discuss and reach a consensus on the matter at hand.¹⁵

2. Substantive issue

The main issue addressed by the Software Interactions Document revolves around the following question: does a specific form of software interaction or interoperation create a work that, if and when distributed, must be so under the copyleft provisions of GPLv2 (and when does it not)?

This question has arisen for two main reasons: first, there is no clear-cut answer to what is a “derivative work”, as defined by copyright law, or “work based on another” (and even if there were, this could vary according to jurisdiction) and second, GPLv2 itself is not clear (or rather, has multiple definitions) regarding what it considers falls within the copyleft obligations of redistribution of the whole under the terms of the GPLv2 (Art. 2b in particular).

The Free Software Foundation, drafter of GPLv2, gives its view on the issue in the GPL-FAQs.¹⁶

You have a GPL'd program that I'd like to link with my code to build a proprietary program. Does the fact that I link with your program mean I have to GPL my program?

Yes.

13 The Berne Convention (last amended 1979) and WIPO Copyright Treaty of 1996 – texts available online at <http://www.wipo.int/treaties/en/ip/berne/trtdocs_wo001.html> and <http://www.wipo.int/treaties/en/ip/wct/trtdocs_wo033.html> respectively. At European level, Council Directive 91/250/EEC of 14 May 1991 on the legal protection of computer programs, available online at <<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31991L0250:EN:HTML>>, consolidated in Directive 2009/24/EC (<<http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2009:111:0016:0022:EN:PDF>>).

14 Discussed in Ravicher, Dan (2002) ‘Software Derivative Work: A Jurisdiction Dependent Determination’, 1, (Nov. 2002), Linux.com, online at <<http://www.linux.com/archive/feature/113252>>. See also Omar Johnny, Marc Miller, Mark Webbink (2010) ‘Copyright in Open Source Software - Understanding the Boundaries’, IFOSSLR Vol 2, Nº1, available online at <<http://www.ifosslr.org/ifosslr/article/view/30>>, DOI: 10.5033/ifosslr.v2i1.30

15 Examples of community debate include: Linux: the GPL and binary modules’ at <<http://kerneltrap.org/node/1735>> and NDIS Wrapper at <http://kerneltrap.org/Linux/NDISwrapper_and_the_GPL>.

16 FSF, GPLv2 FAQs: online at <<http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html>>, section titled ‘Combining work with code released under the GPL’.

While this is a general answer, and the FAQs themselves enter into more discussion on the issue¹⁷ the Software Interactions Document attempts to analyse a subset of these interactions according to the methodology set out below.

2.1. Five Steps

For the purpose of the Document, by way of methodology, the question is broken down into five main questions or steps, the first four looking at copyright law and the fifth looking at additional relevant wording of GPLv2.

1. *What is the original software artefact that is being used in the new work, is it protected by copyright, and to what extent?*
2. *When creating and/or distributing the new work including or interacting with the original software artefact, is any act restricted by copyright being performed in relation to that software artefact, and if so, which?*
3. *Still within the borders of copyright law, if there is no clear-cut answer to these questions, at what additional test or criteria might a court look to determine if the new work could be considered to be the result of the performance of an act restricted by copyright (reproduction or transformation)?*
4. *If you have established that the work in question is protected by copyright, and that the act which you are looking to perform is an act restricted by copyright then, irrespective of any purported grant of licence / permission, does the creation or use of the original work amount to fair use, fair dealing or is any other defense available in the relevant jurisdiction?*
5. *Having done the “bare” copyright-based analysis, set out in the preceding questions, we can finally ask what, if anything, does the wording of the GPL add to this copyright-based analysis (particularly if the answer is in the negative, or at least not clear), and how can that wording be interpreted?*

We look at these questions in turn below:

1. What is the original software artefact that is being used in the new work, is it protected by copyright, and to what extent?

This question raises several issues. The scope of copyright protection is jurisdiction specific. Generally speaking, under the international treaties, works in the public domain and “ideas and principles” underlying the software are not protected (including, under the EU Computer Programs Directive, those that “underlie its interfaces”¹⁸). In the second case (ideas and principles) the scope of these concepts is not clearly defined. US legislation, which excludes any “idea, procedure, process, system, method of operation, concept, principle, or discovery” as mentioned above,¹⁹ and courts (and authors?) seem to have been more active in determining these boundaries, and have

¹⁷ Subsequent FAQs, from <<http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html#LinkingWithGPL>> onwards, e.g.: “What is the difference between “mere aggregation” and “combining two modules into one program?”

¹⁸ EUCPD, Art. 1

¹⁹ US Copyright Act, 17 U.S.C., Section 102(b)

excluded (a) purely functional elements, (b) ideas (when merged with the expression) (c) “scenes a faire”, (d) works in the public domain, and (e) facts, among other limits on copyright protection.²⁰

2. *When creating and/or distributing the new work including or interacting with the original software artefact, is any act restricted by copyright being performed in relation to that software artefact (i.e. is there a clear infringement: reproduction, transformation, distribution?), and if so, which? (merely copying, or copying and transforming?)*

I.e. does creating and redistributing the (combined/inter-related) work involve the performance of an act restricted by copyright, *stricto sensu* (other than distribution of the software artefact itself), regardless of what the GPL may otherwise add. In particular we ask if a particular form of software interaction, under a pure or “bare” copyright law analysis, creates a derivative work of one or both of the interacting software components. This is because the license at least is clear that it applies to derivative works “under copyright law” (here read: strict interpretation of legislation/case law).

Regarding this question, again we find a difference between jurisdictions regarding the creation of a derivative work or “transformation”. While the US law states that a derivative work is a “a work based upon one or more pre-existing works”²¹ (giving rise to tests of substantial similarity and inclusion and a certain amount of interesting case law), the EUCPD talks of “the translation, adaptation, arrangement and any other alteration of a computer program and the reproduction of the results thereof”.²² Within the EU jurisdictions, it seems there is a distinct lack of case law on derivative works of computer programs.

3. *Still within the borders of copyright law, if there is no clear-cut answer to these questions, at what additional test or criteria might a court look to determine if the new work could be considered to be the result of the performance of an act restricted by copyright (reproduction or transformation)?*

This will be even more case specific. In English law, for example, this may be seen within the context of “non-verbatim copying” or similar tests. Here, we could mention, for example:

- *dependency/independency criteria* (does the new work function without the incorporated/inter-related GPL work? Could you swap the GPL component for another one? If so or if not, to what extent? Is there a non-protected API being used as part of the interaction?),
- *“critical functionalities”* (does the GPL component provide critical functionalities for the new work – are these functions more than mere “scenes a faire” or “methods”, that might be excluded under applicable law?),
- *“made for”* (has the plug in been made for a GPL core/kernel, and if so, which part of the core? Does the design of the artefact for which the plug in has been made exert such an influence on the design and development of the plug in that the second developer is (ab)using the skill and judgement of the first?), or

²⁰ Discussed in Samuelson, Pamela (2007) *ibid.*; and Omar, Johnny, et al. *ibid.*

²¹ US Copyright Act, 17 U.S.C., Section 101

²² EUCPD, Article 4

- “use or reproduction of a substantial part of the skill, labour and judgment invested in the original work” when developing the new work.

or other such rationale that (relevant) courts may have used in case law (e.g. copyright is also interested in the manner in which a work is created – which is why there are clean room developments - and not just with which artefact the work interacts or what it does once created, so it could look at the development process).

Note that questions 2 and 3 overlap, or at least it is difficult and even artificial to separate the answers, certainly in case-based jurisdictions where court decisions also establish the law (as opposed to interpreting it).

4. If you have established that the work in question is protected by copyright, and that the act which you are looking to perform is an act restricted by copyright then, irrespective of any purported grant of licence / permission, does the creation or use of the original work amount to fair use, fair dealing or is any other defense available in the relevant jurisdiction?

Again, we meet several challenges as the exemptions from copyright infringement vary from jurisdiction to jurisdiction. In the US one would first look to rely on “fair use” or other explicit exemptions (in the UK “fair dealing” exemptions), while in other EU countries legislation tends to have created a series of specific exemption use-cases, most of which are not relevant for our purposes, but usually include exemptions in favour of interoperability. And there may also be a *de minimis* exception, whereby trivial reproduction will not be covered (in England/Wales, extended by exemption for “insubstantial copying”).

5. Having done the “bare” copyright-based analysis, set out in the preceding questions, we can finally ask what, if anything, does the wording of the GPL add to this copyright-based analysis (particularly if the answer is in the negative, or at least not clear), and how can that wording be interpreted?

We know that with a GPL'd work, there will always be an infringement defence (authorisation) prior to distribution, as the license permits reproducing and transformation... however, an important question is: what are the conditions on exploitation of the third party GPL code (or the plug in for the GPL code)? This is because, for instance, the conditions on copying and distribution are different from those on modifying and distribution. To answer this, we would go back to both the answer to question 2 (which act restricted by copyright is performed?) and the wording of the GPL, and try to resolve any conflicting language.

This is a key question because the GPL purports to cover not only “works based on the Program” as interpreted by copyright law, but also works that “in whole or in part contain ... the Program or any part thereof”, leading us to look into the question of collective/composite works²³ (also possibly considered derivative works- not necessarily the result of an “adaptation/transformation”, but because of the “inclusion”). One may also need to look at the concept of “work” as understood by the GPL (which or what “work” is the GPL talking about?), which is relevant for Art. 2 of the license. For example, a relevant “work” may be a compiled binary which incorporates a GPL'd library. But could a work also be considered to contain a GPL'd library merely because this library

²³ E.g. in Spain, under Articles 8 and 10 of the Spanish Copyright Law RDL 1/1996.

is loaded at run-time? Or the work may interact with GPL'd dependencies (libraries, whatever) that may be distributed separately (or downloaded separately) but are still required by the new work in order to function (disregarding operating system components,²⁴ though even that is a question that must also be answered).

An interesting and valuable view on the concept of “work” is contributed by the FSF itself, when commenting on “mere aggregation” in its GPL FAQs:²⁵

What constitutes combining two parts into one program? This is a legal question, which ultimately judges will decide. We believe that a proper criterion depends both on the mechanism of communication (exec, pipes, rpc, function calls within a shared address space, etc.) and the semantics of the communication (what kinds of information are interchanged).

If the modules are included in the same executable file, they are definitely combined in one program. If modules are designed to run linked together in a shared address space, that almost surely means combining them into one program.

By contrast, pipes, sockets and command-line arguments are communication mechanisms normally used between two separate programs. So when they are used for communication, the modules normally are separate programs. But if the semantics of the communication are intimate enough, exchanging complex internal data structures, that too could be a basis to consider the two parts as combined into a larger program.

So, we ask if the wording of GPLv2, in particular, means that the scope of its copyleft provisions apply to the combined work – whether statically, dynamically linked or otherwise related.

A particular issue with GPLv2 revolves around whether the courts of any jurisdiction would interpret its wording to extend to works which have a connection with a GPL'd work (interact), but are not derivative works *per se* as a matter of law, or their use otherwise does not require the GPL'd code author's consent under copyright law (compilations, collective works, etc.). Thus, possibly enforcing contractual control over the use of the work (which in fact runs against the stated purpose of the license: Article 0 clearly announces: “Activities other than copying, distribution and modification are not covered by this License; they are outside its scope”). The wording of GPLv2 is open to interpretation on this point.²⁶

This issue is best explained by way of example:

²⁴ GPLv2 Clause 3, second separate paragraph.

²⁵ See <<http://www.gnu.org/licenses/old-licenses/gpl-2.0-faq.html#MereAggregation>>.

²⁶ I believe that GPLv3 addresses this point more directly but still not necessarily in a clear manner. Clause 5: “A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an “aggregate” if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate”. Note the exclusion of “independent works” which ARE combined with the GPL'd work such as to form a larger program. See also the definition of Corresponding Source Code (including shared library files) in clause 1.

A coder takes a work subject to GPL2 (“X”), and incorporates a very small part of it (“P”) into another work, (“D”). We have selected P such that incorporation of P in D does not, as a matter of copyright law, require the licence of the original copyright owner of X (this may be because P does not meet the threshold requirements in a particular jurisdiction to attract copyright protection – perhaps it lacks sufficient originality – or because of fair dealing or similar exemptions). It is uncontroversial, that, as a matter of copyright law, the exploitation of D does not require the consent of the copyright owner of X (and follows from our definition of “derivative work”). The question, however, is whether such exploitation of D is a breach of the licence under which X itself is exploited. This issue arises from wording in section 2(b) of GPLv2 which refers to a “work...that...contains...the Program or any part thereof”.

In other words, by distributing a non-derivative work D, may the coder still be in breach of GPLv2 as it applies to X? So, on a licence basis, could the coder potentially lose her licence to X, if this licence purports to require obligations which restrict otherwise-unrestricted acts, and she performs such an act? The answer to this may depend on whether the terms and conditions of the GPL are considered to be a licence or a contract – and if this has been validly formed, etc., but this may not be the only issue. If the answer to this question is “yes”, then we have to look further than copyright law at the relationship between X and D to determine whether the GPL is breached in respect of X, if D is distributed other than under the GPL. If the answer is “no”, then we only have to consider whether D is a derivative work or otherwise covered by X’s copyright rights (in accordance with our definition). So, the question may not necessarily be only “what is one permitted to do by the license in terms of P or D (in our example)?”, but also “how does exploitation of P/D affect the licence for X?”. This subtle distinction should be taken into account when considering each form of software interaction.

2.2. Further comment

At each of these stages, the specific interaction at which we are looking could “fall by the wayside” in GPL v2 copyleft terms, as either copyright protection is not granted, or if it is, there is an exemption, or finally, the license itself provides for exemption from copyleft obligations.²⁷

In addition, we must add that the GPL, as a copyright license, must be interpreted in each jurisdiction under the applicable laws in force (with the additional cross-border complication of determining which law should apply under conflict of law/private international law rules). The US and EU member state laws differ, particularly in their respective formal definitions of “*derivative works*”, “*collective works*” and “*composite/composed works*”.²⁸

And another layer of complication is created if the GPL is considered a contractual document, to which varying jurisdiction-specific rules of contractual interpretation (*contra proferentem*, intention of the parties, etc.) may apply.²⁹

²⁷ For the sake of discussion: a wider interpretation of the license may rely on the use of functional or factual elements of expression of the GPL’d code, which may run against copyright principles which do not protect these parts, even if they are re-incorporated into the new work.

²⁸ The Software Interactions Document provides a legal glossary at the end that discusses these terms.

²⁹ See discussion of GPL as contract, e.g. Moglen, Eben (2001) ‘*Enforcing the GPL*’, online at <http://www.gnu.org/philosophy/enforcing-gpl.html>; Guadamuz, André (2004) ‘*Viral contracts or unenforceable*

3. Example: static linking

As an example of the analysis undertaken, this section presents the interaction mechanism commonly called “*static linking*”.

In static linking, after the source code is compiled into object files, a linker will combine these object files into one executable at build time (“build time linking”, as opposed to “load time linking”). Basically, the linker will copy into the executable the required instructions, data and other symbols of the linked file (and any further object files on which this linked file depends). This one executable will contain the machine code of all the components of the programs that were included in the *link* step.³⁰

An executable is generally considered (by the legal community interested in FOSS) to be a *derivative work* of the programs and libraries contained in the executable – i.e. those that are statically linked into the executable, mainly because this is done by way of reproduction and transformation of those components.

Answering the 5 key questions:

1. The statically linked library is protected by copyright. As a whole, this would include its header information.
2. Copyright in the static library is indeed infringed in the static linking and redistribution of the library through reproduction and arguably modification of the library. The reproduction right is certainly relevant, and arguably the transformation right, it being argued that linking and compiling the library into the executable creates a derivative work of the library (see below). The distribution right is involved, as the library code would be redistributed as part of the executable.
3. Arguably we don’t need to look at any further question, as this software interaction falls clearly within copyright as per answer to Question 1.³¹
4. By incorporating the whole library, no specific exemption may be available (though there have been arguments that even this form of combining is still only “using” the work as contemplated by the author, thus the interaction could be considered fair use (to be read in the light of the GPL, see next questions). Under a free software license, any user has permission to carry out these acts subject to compliance with applicable obligations in the event of distribution.
5. If the library has a copyleft license such as the GPL, the obligations as to redistribution depend on whether one considers that libraries are merely reproduced in the executable or are (also) transformed, as we discuss next.

documents? Contractual validity of copyleft licences’ 26 European Intellectual Property Review 8 331.

30 This is called “*resolving the dependencies*”, by automatic inclusion from external files or libraries in order to satisfy dependencies between the core program and the libraries.

31 See discussion below on wider application of copyright law.

As we have noted, it is generally thought that the process of static linking transforms the library, and thus creates a derivative work. Even if this is not the case, there are also arguments to say that the resulting executable “contains the library” (a collective work or compilation) and thus the executable is a “work based on the program” subject to copyleft obligations or Art. 2 GPLv2 on redistribution. In these circumstances, in order to be permitted to redistribute the GPL library, Art. 2 GPLv2 requires the whole work (the executable) to be redistributed under the GPL.

This is further reinforced by the expressed intent of the GPL “*to control the distribution of derivative or collective works based on the Program*”.³² As an executable with statically linked libraries contains code from those libraries, it is generally thought that this executable should, when distributed, be licensed under the GPL.

Legal appreciations vary.³³

- Whether the original source code of the linking file (i.e. prior to link time) is a derivative work of a statically linked library can be questioned. All it does is reference external required code (symbols, header information of the library), it does not reproduce the linked code in any manner nor does it transform it.
- However, there is an argument that, as the program that contains this source file is designed and written to work with the external library code, it is then dependent on or “based on” the external library (i.e. it is not independent).
- This argument is in turn opposed by a counter argument that the linking program depends more on the interface specifications of the library and/or on the functionalities of the linked library which in both cases are arguably not protected by copyright laws (this may be a stronger argument in the US than in the EU, the US regime excluding “*procedures, processes, systems, methods of operation*” from copyright protection). In this view, creating symbols to refer and thus link to these functionalities is not an act restricted in any way by copyright or, if it were, it should be covered by the doctrine of *fair use*. In addition, it is arguable that – certainly within the EU – the symbols that are used to create the link are themselves protected, being interoperability (interface) information.
- A further argument holds that statically linking the code of a library to create an executable is the expected and normal “use” of a library and thus creating the executable does not entail transforming the library in any way – merely reproducing elements of the library in the executable. Thus the copyleft obligations of Art. 2 GPLv2 do not arise, despite the wording as to “containing the library” as there is no modification (a prior requisite for Art. 2), and rather the obligations under Art 1. (copying and distributing) will apply.

However, despite these arguments, in our experience most lawyers interested in this topic would tend to advise that the fact that the library is statically linked, (i.e code is added to the executable), results in a whole that is derivative of the library.

³² GPLv2, Art. 2.

³³ E.g. see discussion in Rod Dixon (2003) ‘*Open source software law*’, Artech House, at p. 32 et seq. or Välimäki, Mikko (2005) *ibid*; Katz, Andrew (2007) *ibid*.

Finally, we should look at the scope of the concept “Work” in Art. 2: it could cover not just the new code, but the combined work of new code plus original GPL'd software artefact. It is this “whole” that must be distributed under terms compatible with the GPLv2, including all its component parts.

4. Separation and independence

As is set out in the main articles in the Software Interactions Document on the different interaction mechanisms, it is argued that there are some combinations of software programs that will generally always produce a derivative work, while other forms may not. But the dividing line between the two is not clear and in fact will depend on the facts of each case.

For example, one of the major arguments in this area has been that dynamic linking – which does not involve a transformation or compilation/linking of the linked code/library at development or build time - does not necessarily create a derivative work of that code/library, and the external library is only reproduced and distributed (Art. 1, GPLv2), rather than transformed and distributed (Art. 2, GPLv2).³⁴ Although it is then reproduced and linked at run-time (which might create a derivative work), this is only created in the user's computer memory, after redistribution.

If this is correct, the “strong copyleft” view, in order to apply conditions to the distribution of code dynamically linking to the library, may then have to rely on two arguments:

- “collectivity” (for lack of a better word): the dynamically linked library or plug-in is distributed along with the application code that uses it, as an integral part of the “combined” program, and the linking program is not an “independent and separate” work in itself. In this case, the GPL would apply to all the program that is distributed, not just the GPL'd library.
- “interdependency”: the main program that uses the GPL'd library is designed and written to include and use the functionalities of the external library (at runtime) and thus “depends” on the library to work. In this manner, an interdependent compilation has been created, which is argued to fall under the copyleft rules of the GPL.

Neither of these is necessarily a strong or definitive argument, as the new code could be written to a public API and use the GPL'd library as an implementation (among others) of that API. In addition, as we have already mentioned above in respect of statically linked libraries, writing code to use a library (and then executing the library at runtime) could be considered merely “using” the library in the intended manner covered by forms of “fair use”, as well as specifically excluded from the GPLv2 license conditions when it says: “the act of running the Program is not restricted” - thus requiring merely a consent to reproduce (but not modify) and redistribute the artefact. This argument has been set out in the main article of the Software Interactions Document on dynamic linking.

The “dependency” argument is of interest. It has been argued that if the new program is specifically designed and written to work (only) with certain libraries (or vice versa, it is designed

³⁴ Discussed by Dixon Rid (2003), *ibid* at p32 et seq; Katz, Andrew (2007) *ibid*; Rosen, Lawrence (2001), *ibid*.

to be part of an existing third party program, (e.g. like a plug-in), and has little if no other use in any other context), then the program should indeed be considered “based on” (in a contractual meaning, if not a copyright meaning) the third party work. Against this argument, if, in the new work, one could substitute a third party library with another (older, newer, another operating system function, whatever), then it is more likely that the new work would be considered independent of the third party component (and thus either not derivative, or excluded by the “independent” wording of GPLv2).

So in all events the questions of separation, as regards functionalities, design and architecture, etc., and independence between programs / components both at design and development time are relevant questions and, while only based on hypothetical cases provided by our technical colleagues, the Software Interactions Document tries to look at them in each case.

5. Primary and Secondary infringement

Subject to the issues above relating to scope, the Software Interactions Document only considers primary infringement. In other words, potential infringement of copyright by reproduction, transformation (including translation, adaptation and arrangement) and distribution to the public³⁵.

In certain jurisdictions, secondary infringement of copyright is also unlawful (for example, in the United Kingdom, the Copyright, Designs and Patents Act 1988, section 16(2) provides that copyright is infringed by someone who “*without the licence of the copyright owner...authorises another to do...any of the acts restricted by copyright*”. Other jurisdictions have similar provisions). Thanks, by and large, to litigation from rights owners of music and video content who are seeking to prevent the unauthorised distribution of their material by claiming secondary infringement against entities facilitating the unlawful distribution (but who do not themselves distribute – such as holders of peer-to-peer indices) the scope of secondary infringement at law is constantly changing.

It has been argued, for example, that distributing the Linux kernel together with an NDISWrapper amounts to secondary infringement.³⁶ The code of NDISWrapper is released under GPLv2. The argument runs along the lines that, where the Windows XP driver is not available under the GPL, the mere distribution of NDISWrapper somehow authorises a breach of GPLv2 as applicable to the kernel, in that it allows the Windows XP Driver to be interfaced (dynamically, as it happens) to kernel code, and that authorisation of that breach is, therefore, a secondary infringement.

The Software Interactions Document being concerned with primary infringement, no opinion is expressed as to the validity (or otherwise) of that argument. From our example, NDISWrapper itself, in the context of this document, and so far as primary infringement is concerned, can be analysed both from the perspective of a kernel module, and, to the extent that it (potentially) interfaces with non-GPL code, through its interaction with the Windows XP driver by dynamic

³⁵ As set out in Article 4 of EUCPD.

³⁶ An NDISWrapper in this case, is a driver which acts as an interface between the Linux kernel and Microsoft's Windows XP Driver Model interface, and enables hardware for which a Windows XP driver is available, but not a Linux driver, to work with Linux, by using the Windows XP driver instead of the native Windows driver. See debate referred to above, online at <http://kerneltrap.org/Linux/NDISwrapper_and_the_GPL>.

linking. It is argued that *shims* (pieces of code that are themselves typically released under the GPL, but act as an interface to non-GPL code, and of which the NDISWrapper is a specialised example) should be considered similarly.

6. Looking forward

As will be understood from this brief overview of the work that has been carried out on the Software Interactions Document, no definitive answer to the debate has been reached, though hopefully we have provided some useful pointers. Despite the hours of debate and significant number of missives discussing the issue and particular cases, I think we can still safely say that there is no black and white answer, though with luck we have been able to reduce some areas of grey. Uncertainty (from a legal point of view) is one of the major concerns of free software licensing, and we believe that anything that helps to reduce it will be beneficial for the community and industry in general. Having some pointers should help a project weigh up each case and make a more informed decision based on the merits.

As we mentioned above, one of the key objectives of the Document is educational, not just via the discussion of the software interactions covered in the paper, but also via the glossaries at its end, which should provide a common vocabulary on which engineers and lawyers may base their discussions.

As a next stage or step, we can think of two areas of work. First, it would be interesting to take several specific “real world” cases of software interactions (easily done, using publicly available code in any free project), to test the hypotheticals postulated here. Secondly, it would be interesting to expand the analysis, if found useful, with respect to GPLv3 and see what clarification is brought by the more modern wording of this license.

More practically, the Document is a work in progress, and we need more examples and/or diagrams that can help understand the technical issues involved (using header file data, published APIs, etc.) - something that might even be used as a model for presenting and arguing a case either between parties or before the courts.

From a purely legal point of view, bearing in mind the complication of dealing with over 25 jurisdictions (in Europe alone), we believe it would be useful to incorporate further work on the definitions of the legal concepts that are involved, in particular the concept of derivative and collective (composed) works in an IT context. In this respect, the IFOSS Law Book (another project promoted by the FSF-E³⁷) is a useful starter and we look forward to taking advantage of synergies to improve the current work.

³⁷ Online at <http://ifosslawbook.org/>

About the author

Malcolm Bain is partner of Barcelona boutique IP/IT law firm “id law partners” and lecturer in the legal issues of FOSS at the Open University of Catalonia and the University of L rida.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifosslr.org>.

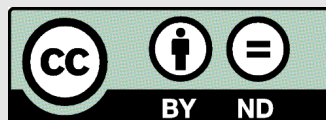
This article should be cited as follows:

Bain, Malcolm (2010) 'Software Interactions and the GNU General Public License', *IFOSS L. Rev*, 2(2), pp 165 – 180
DOI: [10.5033/ifosslr.v2i2.44](https://doi.org/10.5033/ifosslr.v2i2.44)

Copyright   2010 Malcolm Bain.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Affirmative action in procurement for open standards and FLOSS

Mathieu Paapst^a

(a) Teacher/researcher, Center for Law and IT, University of Groningen, the Netherlands.

DOI: [10.5033/ijfosslr.v2i2.41](https://doi.org/10.5033/ijfosslr.v2i2.41)

Abstract

The Dutch action plan Netherlands Open in Connection intends to give a direction for public sector buyers to adopt a positive policy and strategy towards open standards and Free/Libre Open Source Software ("FLOSS"). This policy seems to indicate a preferred position on open source products, however it actually gives no guarantee that providers of FLOSS will get the opportunity to make a competitive offer. For this to be the case there ought to be a 'level playing field' for the FLOSS suppliers and closed source suppliers. However an empirical study carried out between January and June 2010 shows that despite the desired affirmative action for open source products, in almost half (47.5%) of the tenders there is still a preference for closed source vendors or products. This preference inevitably results in not giving vendors of FLOSS a fair chance to win the bid.

Keywords

Public procurement; information technology; Free/Libre and Open Source Software

Info

This item is part of the [Articles](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement. This article has been independently peer-reviewed.

In December 2007 the Dutch government agreed on the action plan Netherlands Open in Connection, hereafter called NOiV, the Dutch acronym. The objectives of this action plan are the following:

1. increase interoperability by accelerating the use of open standards;

2. reduction of supplier-dependence through a faster introduction of open source software and open standards;
3. promotion of a level playing field in the software market, promotion of innovation and of the economy by forceful stimulation of the use of open source software, and by giving preference to open source software during the process of IT acquisition.

To reach these goals the action plan described a number of policies for open standards and open source software. A vast number of these policies directly affected the process of IT procurement within government organisations. A central question for research has been selected: How and under what circumstances does a policy like the action plan NOiV influence behaviour regarding the practice of public tenders?

A subsidiary question is: “Does European procurement law give a fair chance to vendors of FLOSS software or do we need a affirmative action in order to guarantee fair competition?”

Dutch principles and policies

The action plan NOiV describes three principles for the acquisition of FLOSS by organisations in the (semi) public sector.

Principle 1: Open source is not mandatory, but its use should be strongly encouraged.

This principle primarily intends to give a direction for politicians and civil servants to adopt a positive policy and strategy towards open source software. It means that organisations in the (semi) public sector, although not required to choose open source software in all cases, are expected to promote the use of open source software within their organisation. By January 2010 they had to develop an implementation strategy for the procurement, purchase and use of open source software. The political administrative preference for open source software then needed to be translated into that strategy.

Principle 2: Open source software should be preferred if it is equally suitable.

This principle intends to give a direction for government buyers to adopt a positive policy and strategy towards open source software. A possible way to do this is to prescribe weighted award criteria that create a preference for vendor independent solutions. Award criteria in a tender shall be clear and unambiguous and the assessment method must be verifiable, transparent and objective.¹ It is therefore important for public sector buyers to clarify the conditions required for the IT solution in advance in order that suppliers will be put in a position to submit their strongest bids and obtain the highest number of points they can. As long as they relate to the object of the tender the tendering organisation is free to apply various criteria to determine the most economically advantageous offer, including so-called vendor independent award criteria.

The following are examples of vendor independent award criteria:

¹ Directive 2004/18/EC

- The software is database independent and works with databases of several suppliers, including open source databases.
- The user interfaces are browser independent.
- The software can run on more than one operating system (eg. Windows, Linux or Apple).
- Support for the software can be delivered by several (independent) suppliers.
- There are one or more independent user groups or free accessible communities involved in the further development of the software.
- The software uses open specifications and standards.
- The legal terms of use refer to aspects such as availability of the source code and the conditions for modification and distribution of the software.

Determining the weight of such award criteria in terms of percentage could also be included in the implementation strategy document. If an organisation actually prefers supplier independent software they can use a high weighting for these criteria. Although the criteria do not exclude closed source proprietary solutions, they can value a bid using FLOSS and open standards more than an equivalent and equally suitable bid using closed source proprietary software.

Principle 3: Providers of open source software should have the same opportunities as providers of closed source software.

Although the two principles discussed above seem to indicate a preferred position for Free and Open source software at first glance, they actually give no guarantee whatsoever that providers of FLOSS will get the opportunity to make a competitive offer. For this to be the case there ought to be a 'level playing field' for the Free and open source software providers and the closed source software suppliers. The action plan mentions this in the following way:

"The Cabinet wants procurement and purchasing processes of software to give the same opportunities to providers of open source software (...)." ²

This is a surprising admission by the Dutch government: that the current European procurement laws and procedures have not yet accomplished the creation of this level playing field in the software market and are therefore, one could argue, not realizing their initial goal which is stimulating competition in order to complete the European single market.

Dutch vendors of proprietary closed source software however argued that the preference of the Dutch government for Free and Open source software is in fact a violation of European procurement laws.³

² Actionplan Netherlands Open in Connection, p.17

http://www.whitehouse.gov/files/documents/ostp/opengov_inbox/nl-in-open-connection.pdf

³ See e.g.: <http://www.channelweb.nl/nieuws.jsp?id=3196657> (Dutch)

Research

To see if European procurement law does give a fair chance to vendors of FLOSS empirical quantitative research was carried out which asked for the data of 94 Dutch calls for tender, published between January and June 2010, that followed the open procedure and that consisted of the delivery of software of some kind.⁴ Out of the total sample of 94 calls, data relating to 80 tenders was received, a response rate of 85,1%.

All these tender documents were examined on 25 different aspects, such as the needs or wants for open standards, vendor independent award criteria, the possibility to use ODF (a document format based on an open standard) for the bid and the possible preference for open or closed source products.

Results

The following results emerged:

In 36 cases (45%) the tender documents mentioned a need or want for open standards in general. In the remaining 44 cases open standards were not requested. This is not a very positive result considering the fact that it has been Dutch policy to ask for open standards since 2002 when the Dutch Parliament adopted the motion “Vendrik.”⁵ Under that motion the government was asked to ensure that by 2006 all public sector software used open standards. To reach this ambitious goal it would seem necessary to ask for open standards in all public sector tenders.

Request for open standards	Frequency	Percent
Yes	36	45
No	44	55

Table 1: Frequency of needs and wants for open standards

The possibility of a vendor using the odt (ODF) format for his bid was considered. According to the action plan all the ministries and subsidiary government bodies should be able to receive documents in the ODF open standard by January 2009 at the latest.⁶

In almost half of the cases (45%) the use of ODF was possible. In the other 46.2% it was not

⁴ The tenders that asked for Voice over IP technologies or printer hardware with printer drivers were not included in the sample of 95 tenders due to technical expertise limitations.

⁵ Lower Chamber, 2002/2003, 28 600 XIII, nr. 30 (Dutch) The motion, named after its initiator Kees Vendrik, who is a former member of Dutch parliament, also asked for ambitious steps to ensure the preference and stimulation of FLOSS.

⁶ Actionplan Netherlands Open in Connection, p.9 http://www.whitehouse.gov/files/documents/ostp/opengov_inbox/nl-in-open-connection.pdf

possible, however this was mainly caused because the tendering organisation demanded the use of the PDF format. Only in one case both PDF and ODF were not possible because a vendor was obliged to use a Microsoft Word and excel format suitable for Windows XP.

In the 7 remaining cases (8.8%) a digital bid was not requested by the tendering organisation.

Possibility of using ODF	Frequency	Percent
Yes	36	45
No	37	46.2
n/a	7	8.8

Table 2: Frequency of possibilities to deliver the bid in ODF

Looking at the possible presence of a preference for FLOSS, in 22 cases (27.6%) a preference for FLOSS, or more particularly “Open source licensing”, was mentioned. This was mostly done by mentioning the action plan NOiV as a principal guideline for the procurement of software or by actually mentioning that the customer preferred open source software. One should keep in mind that this is not the same as a preference for a named open source product or vendor. At first glance 27.6% seems to be a positive result, however in only 15 cases (18.8%) the preference actually resulted in a reward of extra points under the weightings applied to the award criteria. This means that in 7 cases (8.8%) a preference has been mentioned, but a vendor would not get extra points for offering a FLOSS product. In 1 case there was a preference for FLOSS combined with a clear preference for a named enterprise edition “open source” product. For the purposes of this survey, this was counted as a preference for FLOSS with reward of extra points. However, it is highly debatable whether an “enterprise edition” can fulfil the same goals as a community edition whether it helps to reach the NOiV objective of reducing the dependence on suppliers. There is a strong argument that buying an enterprise edition which is not being delivered under an OSI approved license (or something broadly similar) is basically the same as a closed source proprietary product, having the same disadvantages.

Preference for FLOSS	Frequency	Percent
FLOSS preferred and rewarded	15	18.8
FLOSS preferred but not rewarded	7	8.8
FLOSS not preferred	58	72.4

Table 3: Preference for FLOSS

The tender documents were examined for a preference for closed source software, and in particular a preference for a named closed source product or vendor. The mere use of a trademark or product name in public procurement (which is actually a widespread practice)⁷ was not, by itself, considered sufficient to demonstrate such a preference. In lots of cases trademarks and product names are used to describe both the current architecture and the software the new solution has to integrate with.⁸ For the purpose of this study, such a use of trademarks and product names was not taken to have established a clear preference for a product or vendor, although one could argue that it becomes a discriminating preference the moment compatibility is required with previously purchased proprietary software, especially if the technical specifications needed for that compatibility are not publicly available and freely usable.⁹

For the purposes of this study, actual discriminatory use of trademarks, patents, types, and legal and technical conditions in relation to the vendor or product which was the subject of the procurement needed to be present in order to establish a preference for closed source vendors or products. In 29 cases (36.2%) a clear preference for a named closed source product or a closed source vendor was found. Accordingly in these 29 cases other vendors than the preferred one did not have a fair chance to win a bid.

7 See e.g. OpenForum Europe, 2008. "OFE Monitoring Report: Discrimination in Public Procurement Procedures for Computer Software in the EU Member States", where 34 out of 136 tender notices mention trademarks in procurement documents. <http://www.openforumeurope.org/press-room/press-releases/procurement-study-release.pdf>

8 According to Rishab Gosh et al, "Guidelines on Public Procurement of Open Source Software", this might not be a legitimate functional requirement according to article 23 (8) of the Directive 2004/18/EC since software can usually be described in terms of standards and functionality.

9 In decision T-345/03 of 12/03/2008 the Court of first instance of the European Community considers that the Commission infringed the principle of equal treatment as between tenderers by failing to make available to all the prospective tenderers from the beginning of the tendering procedure the documentation relating to the technical architecture and source code and that that infringement could thus have affected the award of the contested contract. <http://curia.europa.eu/>

Preference for closed source vendor or product	Frequency	Percent
Yes	29	36.3
No	51	63.7

Table 4: Preference closed source vendor or product

In practice there are many examples of discriminatory criteria used to give a preferred position to named closed source proprietary software or vendors, including:

1. Asking for a named product or licence of a named product (eg. tender for Windows licenses)
2. Asking for a reseller of either licences of the closed source installed base or new named closed source software.
3. The vendor must be a certified and recognised partner of a named closed source product.
4. The vendor must have an agreement with Microsoft for technical support on the product.
5. The product must be certified by a named closed source vendor.
6. The vendor must deliver a new product and have permission to expand the closed source installed base.
7. The vendor must employ Microsoft certified employees.
8. Asking for an operating system that can be downgraded to Windows XP.
9. Asking for an operating system that can be used together with the Microsoft Campus Agreement.
10. Tendering for hardware (e.g. laptops) while also asking for named software (in most cases Microsoft) to be installed on that hardware.

In two of these 29 cases the tendering organisations actually mentioned that they had a preference for a named closed source product and vendor.

Finally some of the other criteria that could prevent vendors, and in particular FLOSS vendors, from making a bid and having a fair chance of winning were considered. In 9 other cases restrictions were found that made it very difficult or impossible for vendors to offer a FLOSS product.

Some examples of these criteria are the following:

1. If your bid is open source you should give extra guarantees concerning the stability of the open source community.

2. The vendor has to be the copyright owner.
3. Extra points are to be awarded if interoperability with a named closed source product is already integrated within the offered product.
4. Not allowing licences to be offered for a “zero-price”.
5. Demanding that offered applications must be certified by Microsoft, are Oracle 10 compliant and use the official Microsoft style guide as much as possible.

Restrictions preventing fair competition	Frequency	Percent
Yes	38	47.5
No	42	52.5

Table 5: Frequencies of restrictions for FLOSS vendors

This shows that despite the desired affirmative action for Free/Libre and Open Source Software in almost half (47.5%) of the sampled tenders there was still a preference for closed source vendors or products. This preference inevitably results in vendors of FLOSS products not receiving a fair chance to win the bid.

The last question to be considered was whether or not it would make any difference to the prevalence of these restrictions if a tender was organised completely by the tendering organisation itself or if the procurement process was “outsourced” to an external advisor or company. A reasonable hypothesis would be that the external advisors, who are presumably all experts in the field of procurement, would do better in following the EU directives. After comparing the numbers this hypothesis proves to be wrong. In 36.7% of the sixty internally guided procedures there were restrictions on FLOSS vendors or products. In the twenty externally guided procedures however, these restrictions were found in 80% of the cases.

Restrictions preventing fair competition	Internal (N=60)	External (N=20)
Yes	22 (36.7%)	16 (80%)
No	38 (63.3%)	4 (20%)

Table 6: Frequencies of restrictions in externally and internally guided procedures

Conclusion

This research shows that despite the desired affirmative action for Free/Libre and Open Source Software, in almost half (47.5%) of the tenders there is a preference for closed source vendors or products. Because of this preference vendors of FLOSS products are not given a fair chance to win the bid. There is no level playing field on the software market and government buyers arguably do not act according to the EU treaty principles of equal treatment, non-discrimination and transparency. Further research will try to determine why public sector buyers disregard the EU directives and regulations and how FLOSS vendors experience this behaviour.

About the author

Mathieu Paapst holds a masters degree in Law and IT and works as a lecturer and PhD-researcher at the Center for Law and IT, University of Groningen. He is interested in legal and social aspects of IT procurement, copyright licensing and e-Health, and has been publishing about open standards and free software since 2005.

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifoisslr.org>.

This article should be cited as follows:

Paapst, Mathieu (2010) 'Affirmative action in procurement for open standards and FLOSS', *IFOSS L. Rev.*, 2(2), pp 181 – 190

DOI: [10.5033/ifoisslr.v2i2.41](https://doi.org/10.5033/ifoisslr.v2i2.41)

Copyright © 2010 Mathieu Paapst.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Software Package Data Exchange (SPDX™) Specification

Kate Stewart,^a Phil Odenice,^b Esteban Rockett^c

*(a) Ubuntu Release Manager, Canonical, Inc.; (b) Senior Vice
President of Business Development, Black Duck Software; (c)
Senior Counsel, Motorola Mobility, Inc.*

DOI: [10.5033/iffosslr.v2i2.45](https://doi.org/10.5033/iffosslr.v2i2.45)

Abstract

The goal of the Software Package Data Exchange (SPDX™) specification is to enable companies and organizations to share license and component information (metadata) for a software package and related content with the aim of facilitating license and other policy compliance. The specification is being developed through collaboration between technical, business and legal professionals from a range of organizations to create a standard that addresses the needs of various participants in the software supply chain.

Keywords

License Compliance; Copyright Identification; Specification Format; Software Package Metadata; Software Supply Chain.

Info

This item is part of the [Tech Watch](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement. This article has been independently peer-reviewed.

Background

Companies at all points in the software supply chain are becoming conscious of the need to treat open source just like any other third party code. They need to know and document the components in the products and software they are consuming and distributing. There are a variety of reasons for this, not the least of which is to make sure they understand their legal obligations. Thus the need for a common approach to sharing information about software packages and their related content has never been greater. Breaking down information silos is still a work in progress. The Software Package Data Exchange working group¹ was formed originally as a FOSSBazaar

¹ <http://www.spdx.org/>

sponsored effort² and is now a part of the Linux Foundation's Open Compliance Program³. The working group's goal is to define a way to share copyright and license information about a software packages and common licenses in that package, down to the file level.

Why is a standardized specification needed?

Innovation happens very rapidly in the open source ecosystem, often by developers by building on top of the work of other developers. To do this, source code files, that have been created as part of one project under a specific license may be copied and reused in another project that may be under a different license. This mixing and matching of licenses, creates problems for those companies reusing and redistributing software packages that contain this combined software. It becomes a real challenge for them to figure out what they need to do to comply with the licenses that govern their software packages. By creating a standard way of summarizing the licensing and copyright information to the file level and providing a way to double check that the summary actually matches the code, a standard makes the task of figuring out what license are in effect much easier. This permits creation of a software "bill of materials" that can be passed with the actual software, throughout the supply chain, saving considerable analysis effort at every step. Simply saying your company is doing the right thing is not enough: savvy consumers in the supply chain want proof to limit the risk of non-compliance with licenses. Suppliers themselves welcome a single standard format for disclosing open source rather than having to respond to each customer's request using a unique format.

What does the SPDX™ specification consist of?

The SPDX™ effort has focused on coming up with a way to summarize the discoverable facts about code content and ownership. By providing a 'defined format of file to accompany any software package,' the effort eases the burden of exchange of license information between companies. The standard defines a format for sharing: facts that deal with identification, facts that provide overview information, and facts that provide file-specific information about the software package.

Facts that deal with a software package's identification (metadata) included in the specification are:

- Version of the SPDX™ specification is in use
- Unique identifier (based on a cryptographic hash algorithm) representing a unique identifier that correlates with this specific software package
- Method by which information was generated (who, when, tools used, etc.)
- Independent audit information (sign-off/reviewed by)

Facts that provide overview information about a software package's content include:

² <https://fossbazaar.org/>

³ <http://www.linuxfoundation.org/programs/legal/compliance>

- Formal Name
- Package File Name
- Download Location
- Declared License(s)
- Detected License(s)
- Copyrights and Dates

Facts that are specific to a software package's file-specific properties:

- File Name (including subdirectory)
- File Type (source or binary)
- Detected license(s) governing file (from file)
- Copyright owners and dates (if listed)

Because of the license orientation of the specification, the working group is also committed to providing standardized license references. The specification includes:

- License names
- Unique identifiers for common open source licenses
- Mechanisms for handling non-standard licenses.

The SPDX™ specification does not attempt to transmit legal judgement, but rather provides a format for a summary of the facts from which professionals (perhaps using other tools) may make judgements.

How far along is the development and what are the next steps?

The Version 1.0 beta form of the specification is available for download⁴, but it is just a starting point. It has had some road testing, but has not been driven by the public, so the group's focus is shifting to driving practical applications and incorporating the inevitable feedback before we release the official version 1.0. The group is assembling a list of key projects for which to create SPDX™ reports, and to get create those reports by any method possible. Initially we expect members of the group to roll up their sleeves on this live testing, but we are also working hard with tool vendors (proprietary and open source) to create other options for generating these reports. We anticipate the need to develop new tools (e.g. syntax checkers and reading and displaying tools) to enable this development, as well as training materials for educating others on using the standard.

⁴ <http://www.spdx.org/spec/current>

Interested in learning more and helping out?

If you want to join our volunteer effort, and help make it better, there is information on how to participate available on our web site⁵. Sub-groups with their own mail lists have recently formed around technical, business, and legal issues, and depending where your interests are, all are open and welcome new members to collaborate on the specific topic areas.

Conclusion

Getting the SPDX™ specification adopted across the ecosystem will be a challenge. We need participation and support from key Linux distros and package maintainers, legal experts, tool developers (commercial and open source) and package consuming organizations as well. With major players in all those categories already on board, and with the support from FOSSbazaar and the Linux Foundation, the pieces are finally coming together to let us achieve our goal of a useful specification.

About the authors

Kate Stewart is Ubuntu's Release Manager at Canonical, Inc. After reviewing way too many standard projects for license and copyright info in her prior job at Freescale Semiconductor, Inc., she found a group of folk equally frustrated, and set out collaborating with them on defining a specification for sharing package licensing and copyright facts between projects and organizations.

Phil Odence is Vice President of Business Development for Black Duck Software, makers of enterprise application development tools that address management, compliance and security challenges associated with open source. In that role, he is responsible for expanding Black Duck's reach, image and product breadth by developing partnerships in the multi-source development, legal and open source ecosystem.

Esteban Rockett is a Senior Counsel at Motorola Mobility, Inc. He is Mobility's lead software intellectual property counsel, including serving as legal lead for its open source contributions and compliance, mobile application stores, and digital content licensing and delivery.

5 <http://www.spdx.org/wiki/spdx/participation-guidelines>

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Stewart, K., Odence P., Rockett, E. (2010) 'Software Package Data Exchange (SPDX™) Specification', *IFOSS L. Rev.*, 2(2), pp 191 – 196

DOI: [10.5033/ifosslr.v2i2.45](https://doi.org/10.5033/ifosslr.v2i2.45)

Copyright © 2010 Kate Stewart, Phil Odence, Esteban Rockett.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Never a Better Time for Open Source

Matt Asay,

DOI: [10.5033/ifosslr.v2i2.48](https://doi.org/10.5033/ifosslr.v2i2.48)

Abstract

Matt Asay, senior vice president of business development at Strobe, takes stock of developments in the legal environment surrounding open source which have taken place in 2010.

Keywords

Law; information technology; Free and Open Source Software

Info

This item is part of the [Tech Watch](#) section of IFOSS L. Rev. For more information, please consult the relevant section policies statement.

If we needed any further confirmation that open source has gone mainstream, just look at the number of lawyers circling its skies. The European Legal Network, one of if not the main legal network, now boasts over 250 lawyers and of course in the last two years, spawned this journal.

No one bothers to sue the impoverished: now that open-source software is minting money for Google, IBM, Red Hat, and others, it has taken centre stage in the litigation front, through 2010.

While many of us would probably prefer to remain off the legal radar, the reality is that legal action is part and parcel with open source's importance in the industry. Bad as it is to be constantly fending off patent lawsuits, the opposite is even worse.

Why is open source so relevant to the technology industry today? Because we are in the midst of a tectonic shift in how software is delivered; a shift from value in the bits themselves to value in the services around the bits. Cloud computing, Software-as-a-Service, and open source each presents different faces on this trend. Small wonder, then, that Dresdner RCM Global Investors principal Walter Price, Jr. anticipates "a secular decline in tech spending," pointing the industry "to an environment where it will be difficult for [tech vendors] to keep revenues growing."¹

As incumbent vendors struggle to remain relevant, one response has been to launch lawsuits to slow their obsolescence. Like the water-seekers in T.S. Eliot's epic poem "The Wasteland", who

¹ <http://online.barrons.com/article/SB50001424053111904502004575562243330821352.html>

find "no water but only rock," leaving them with with "red sullen faces [that] sneer and snarl."

And sue, apparently.

In 2010, Android was the most obvious recipient of the legal profession's attentions, with Apple, Microsoft, Oracle, and many others lining up to halt the open-source platform's rise or at least profit therefrom. Rather than sue Google directly, most of these parties preferred to sue different members of the Android ecosystem like HTC, the handset manufacturer.

While Google has not publicly commented on many of the suits, given that it is not directly named in them, Google did vigorously rebut Oracle's claims of patent infringement for the way Google uses Java in Android. Interestingly, its response was less about the law than about the politics and history of Oracle's contradictory and not-so-open approach to managing Java.² In responding to Oracle's suit, Google attempted to rally the open-source community to its side:

"We are disappointed Oracle has chosen to attack both Google and the open-source Java community with this baseless lawsuit. The open-source Java community goes beyond any one corporation and works every day to make the web a better place. We will strongly defend open-source standards and will continue to work with the industry to develop the Android platform."

Good politics, but not necessarily sound legal doctrine.

Perhaps Google was hoping that public opinion could gain it some clarity in the dispute, given that the law increasingly does not. In 2008 many of us saw the *Bilski* case as a ray of hope, a sign that U.S. courts would invalidate decades of bad law that tried to shape intellectual property to mirror the world of physical property, with adverse effects.³ Since that time, however, no industry-wide decimation of the patent regime has occurred, either in the United States or in Europe, and it is looking doubtful that it ever will. The poor, and patents, are evidently always with us.

With this in mind, it strikes me that organisations like the Open Invention Network and other open-source friendly patent collectives will gain an increasing importance in the industry. Most of the upstarts giving established vendors fits lack significant patent portfolios, making a collective approach the only feasible manner of competing with the SAPs and Oracles of the world on somewhat equal footing.

I also believe that the open-source legal community needs to continue to lower the bar to adoption of and participation in open source. One current endeavour is Project Harmony, an attempt to systematise contributor agreements for open-source communities. Despite the opposition from some in the broad open-source community for such a project, it can do much to help would-be participants navigate the complexities of open source, similar to what the Open Source Initiative has done for normalising open-source licensing.

There has never been a better time for open-source software, and the increasing frequency of intellectual property claims against prominent users and developers of open source is one clear

² <http://www.scribd.com/doc/38782601/Google-Answer-to-Oracle-Java-lawsuit>

³ <http://www.reuters.com/article/idUSTRE49T7HI20081031>

sign of this. Open source increasingly fuels the growth of companies like Facebook and Twitter, while simultaneously challenging the standard operating procedures of our industry's venerable (and more and more vulnerable) incumbents. Game on.

About the author

***Matt Asay** is senior vice president of business development at Strobe, a startup that offers an open source framework for building mobile apps. He was formerly chief operating officer of Ubuntu commercial operation Canonical. With more than a decade spent in open source, Asay served as Alfresco's general manager for the Americas and vice president of business development, and he helped put Novell on its open-source track. Asay is an emeritus board member of the Open Source Initiative (OSI). His column, *Open...and Shut*, appears twice a week on *The Register*.*

Licence and Attribution

This paper was published in the International Free and Open Source Software Law Review, Volume 2, Issue 2 (December 2010). It originally appeared online at <http://www.ifosslr.org>.

This article should be cited as follows:

Asay, Matt (2010) 'Never a Better Time for Open Source', IFOSS L. Rev., 2(2), pp 197 – 200

DOI: [10.5033/ifosslr.v2i2.48](https://doi.org/10.5033/ifosslr.v2i2.48)

Copyright © 2010 Matt Asay.

This article is licensed under a Creative Commons UK (England and Wales) 2.0 licence, no derivative works, attribution, CC-BY-ND.

As a special exception, the author expressly permits faithful translations of the entire document into any language, provided that the resulting translation (which may include an attribution to the translator) is shared alike. This paragraph is part of the paper, and must be included when copying or translating the paper.



Volume 02 }

Issue 02 } December 2010

Thanks to our Sponsors

IFOSS L. Rev. is published by its Editorial Committee.
The Committee gratefully acknowledges the sponsorship
of its many supporters and donors, including those who
have permitted the upstart of the review with their
contribution :



Available online at: <http://www.ifoisslr.org>