Collaborative Software Development: Who Owns Copyrights?

Iryna Lishchuk
Institut für Rechtsinformatik, Leibniz Universität Hannover
L3S Research Center
Hannover, Germany
email: iryna.lishchuk@iri.uni-hannover.de

***Abstract*—This paper investigates the nature of collaborative research projects in terms of software copyright and data ownership. It discusses and seeks to answer such questions as: Who owns copyright in collaborative software development? If several partners contribute into the software toolkit, how do they share copyright? If data driven software is developed and trained against personal data, does it affect copyright ownership? How the data providers and developers share the rights? The legal analysis is conducted against research action undertaken in the medical research project HarmonicSS, supported by case studies from open source projects.**

Keywords-copyright ownership; data sharing; data driven software; sharing of rights.

I. INTRODUCTION

In research projects, software developments are normally a result of collaborative work. Sooner or later, a question: "Who owns copyright?" arises. Is it a software developer who carried out the work that owns software copyright or is it a partner institution? May a software developer decide on release of his software developments open source? When a number of partners develop a software work jointly or contribute individual modules, how do the partners share their rights? The matter of copyright ownership is important for exploitation. First, it is the holder of copyright, who has the right to exploit the work. Second, it is also the right holder, who has the power to decide the licensing strategy.

The matter becomes even more complicated when data driven software is developed in collaboration with data providers. In this situation, apart from software developers also the data providers come into play. The logic question arises: if software is developed against the data and rights in data belong to the data providers, who owns rights in data driven software: data providers or developers? An associated concern on part of data providers is whether their data rights are affected by the software development process.

The author considered the licensing implications of "open source" software elsewhere [1]. The critical issues behind data sharing have also been well articulated [2]. The focus of this paper is on the sharing and management of copyrights and data ownership in collaborative research projects. The research project HarmonicSS is a good example for this.

HarmonicSS is a large-scale ICT medical research project in the domain of personalized medicine [3]. Full title is "*HARMONIzation and integrative analysis of regional, national and international Cohorts on primary Sjögren's Syndrome (pSS) towards improved stratification, treatment and health policy making disease*". The HarmonicSS vision is to create and maintain a platform with open standards and tools entrusted to address the unmet needs in primary Sjogren Syndrome (pSS) and designed to enable secure storage, governance, analytics, access control and controlled sharing of information at multiple levels. The research work is done in collaboration and a number of project results are developed by multiple institutions, including technical experts and clinical partners. An example is Patient selection tool for multinational clinical trials. The tool is aimed to select patients from the integrative cohort eligible for multinational clinical trials for new pSS treatments. The technical background is composed by a Service Oriented Architecture (SOA) and open source tools and models developed in the PONTE project: Efficient Patient Recruitment for Innovative Clinical Trials of Existing Drugs to other Indications [4]. The models include: Clinical Trial Protocol Authoring Tool, Eligibility Criteria Model, Set of mechanisms and models linking to healthcare patient data sources for clinical research querying, Decision Support during study design and patient selection, etc. Apart from the data protection issues, which such collaborative medical research calls into play, the issues of data ownership and software copyright are not less essential. Several factors matter here, namely: Who owns copyright if several contributors are involved? How co-owners share the copyright? How the works developed in collaboration can be exploited and what are the pre-requisites for that? Are there any implications produced by data driven software development for the rights in data? Who owns the results? The legal implications behind the copyright and data ownership issues and potential options how such issues may be resolved we consider next.

The rest of this paper is organized as follows. The doctrine of first ownership in copyright is discussed in Section II. The nature of software development projects in terms of copyright is considered in Section III. Section IV elaborates on data rights. The management of copyrights by

 ISBN: 978-1-61208-655-2

contractual means are discussed in Section V. Conclusions finalize the paper.

II. FIRST OWNERSHIP

This Section elaborates on the principle of first ownership in copyright both from legal background and practical implications.

A. Legal Background

This study relates to the field of copyright law and examines the legal relations in collaborative software development across jurisdictions. However, the cross-border nature of collaborative software development does not change the legal background much. The copyright law is at much extent harmonized across jurisdictions. The main legal instruments of copyright law, such as the Berne Convention, the WIPO Copyright Treaty and the TRIPS Agreement introduce the minimum standard of copyright that shall be implemented by all Member States to the WTO and the Berne Union. The high-level study of copyright law suffices to examine issues discussed in this paper. The focus is made on harmonization of copyright in the EU, particular in the field of software copyright, such as introduced by the Directive 2009/24/EC on the legal protection of computer programs (Software Directive) [7].

To start with, it may be beneficial to note that all open source licenses, both as proprietary licenses, start with copyright notice, namely the declaration about who owns copyright. The copyright mark © denominates who holds software copyright in a program and has the right to dictate software distribution or licensing in one or another way. The copyright line, as integrated into the Apache License [5] looks as follows:

"Copyright [yyyy] [name of copyright owner]"

However, it is a typical situation in software development that the programmer who writes the code is the author of a program, but not necessarily the copyright holder. The code means a source code, written in one or another programming language; whereas software licensed "open source" must include source code, and must allow distribution in source code as well as compiled form.

According to the rule of first ownership in copyright, "*the first owner of copyright in a work is usually the author of the work"* [6]. It means, copyright in a work inherits the creator of a work, i.e. the author – a natural person. The same principle applies in software copyright. According to Article 2 (1) Software Directive:

"*The author of a computer program shall be the natural person or group of natural persons who has created the program or, where the legislation of the Member State permits, the legal person designated as the rightholder by that legislation."*

However, under the work-for-hire doctrine, copyright in a work, created by an employee in course of employment, passes to the employer. This principle has also been anchored in software copyright and is reflected in Article 2 (3) Software Directive:

"*Where a computer program is created by an employee in the execution of his duties or following the instructions given by his employer, the employer exclusively shall be entitled to exercise all economic rights in the program so created, unless otherwise provided by contract.*"

In this constellation, namely where a computer program is created by a developer under employment, the developer bears moral rights in a program he creates, such as the right to be named as the author, whereas the employer inherits the economic rights. The moral and economic rights constitute full-fledged copyright. The moral rights are inalienable by nature and reserved by the author at any time. The moral rights are recognized by Article 6bis (1) Berne Convention:

"*Independently of the author's economic rights, and even after the transfer of the said rights, the author shall have the right to claim authorship of the work and to object to any distortion, mutilation or other modification of, or other derogatory action in relation to, the said work, which would be prejudicial to his honor or reputation."* [8]

The economic rights encompass entitlements to the commercial exploitation of a work. The basic economic rights include the right to distribution, reproduction, modification and making available to the public [8].

In application to collaborative software development, this principle means the following. When a software developer writes a program for the project in the status of an employee, then, in the absence of an agreement, the employer, namely the partner institution holds software copyright. Consequently, it is the partner institution that has the legal position of the right holder and has the power to decide on the licensing strategy (be it open source or proprietary).

In a situation, where a programmer writes a program acting as a freelancer or subcontractor, then according to the rule of first ownership in copyright, it is the software developer who owns copyright and holds all moral and economic rights in a program, unless contractually agreed otherwise [6]. The difference between a freelancer and employee is that a freelancer sells his services to the employer without a long-standing commitment. By contrast, an employed developer commits to provide software development services to the employer under certain conditions for a specific period of time in return for remuneration. For example, the parties (the customer and developer) may agree that all economic rights in a program are assigned to the customer. This being the case, the customer is entitled to decide on the licensing software "open source". By contrast, if only use-license is negotiated, allowing the customer to run the program for his needs, the economic rights stay by the developer.

At the same time, provided a programmer writes the code and/or contributes into an open source project in his spare time, the rule of first ownership in copyright applies and it is the programmer who owns both moral and economic rights in the program he creates [7].

B. Practical Implications

In fact, the issue of copyright ownership plays an important role in software exploitation. As the case law shows, the ignorance and/or disregard to the issue of

copyright ownership, namely who holds copyright in a code: employer, software project or the developer often leads to copyright litigation. It may be observed, that often in cases where popular IT companies litigate over a piece of software, which company B allegedly copied from company A, the dispute often arises from the fact that company B hired a developer X from company A, who wrote the piece of code at issue and integrated that piece of code into a software product of company B.

One example is the case Oracle America, Inc., v Google Inc. [9], tried by the U.S. courts from 2012 through 2016. In principle, the case concerned copyrightability of Java APIs, namely whether the Java APIs are protected by copyright. In brief, the copyright in Java APIs was recognized in the appellate instance [10], followed by the Google fair use defence and petition to the U.S. Supreme Court to review the case. Finally, the dispute was decided in favor of Google with verdict recognizing Google re-implementation of Java APIs as fair use [11].

Apart from the API copyrightability issue, there was also a small piece of code, which Orcanle claimed was replicated from Java into Android verbatim. And that piece of code made its way into Android in the result of Google hiring software engineer from Sun.

Dr. Joshua Bloch worked at Sun as a distinguished engineer specializing in Java from August 1996 through July 2004. While at Sun, Dr. Bloch wrote the nine line code called "rangeCheck". It performed a function to check the range of values before sorting the list. This function was put into a file, "Arrays.java", which was part of the class library for the 37 API packages at issue. In 2004, Dr. Bloch came to work to Google. In his spare time, he continued working on Java, and around 2007 wrote the files "Timsort.java" and "ComparableTimsort". These files also contained the same "rangeCheck" function that he wrote while at Oracle before. Dr. Bloch contributed his Timsort file to OpenJDK and Sun included Timsort as part of Java J2SE 5.0 release. In 2009, while working on Google Android project, Dr. Bloch contributed Timsort and Comparable Timsort to the Android platform. And this is how the nine line „rangeCheck" happened to be in Android and this was how the infringement happened to occur [9].

When discovered, the „rangeCheck" was removed from the Android edition. Because „rangeCheck" was nine lines appearing in a class of 3,179 lines of code, it was found as "an innocent and inconsequential instance of copying in the context of a massive number of lines of code" [9]. This example demonstrates how the constellation and the legal relations, in which the programmer has written the code, may affect copyright ownership and produce some legal implications.

III. NATURE OF SOFTWARE DEVELOPMENT IN TERMS OF COPYRIGHT

This Section considers the works of collaborative software development in terms of copyright: joint works, composite works and derivative works.

A. Collaborative Nature of Software Projects

The collaborative nature of software development process brings another copyright relevant issue into play, namely: "How all the contributors share copyright?"

As noted above, software development projects are normally collaborative projects, which receive contributions from a number of software developers, who often work and contribute their bits of code independently [12]. As a rule, such collaboration results in a software product combined from inputs of various contributors. Thus, a large number of people may be involved in initial development, but even more can work on revised versions and updates [6]. As one author commented: "*Given the growing expanse of users working collaboratively, today's Linux is less a seamless piece of coding than a tapestry of hundreds of hackers' contributions."* [13].

However, in legal terms, a ´derivative work´, a ´work of joint ownership´ and a ´composite work´ shall be distinguished. The legal consequences that these three formats produce vary.

B. Joint Work

The legal nature of a joint work reflects the idea of co-authorship [12]. The UK Copyright Act CDPA 1988, Section 10 (1), defines a work of joint ownership as "*a work produced by the collaboration of two or more distinct authors in which the contribution of each author is not distinct from that of the other author or authors"* [14]. The main characteristic of a joint work is that contributions are not separable, are not distinct from each other and do not constitute separate works in themselves and cannot be protected in their own right. Another essential factor, which marks a joint work, is intent of the contributors for their inputs "*be merged into inseparable or interdependent parts of a unitary whole*." [12]. In other words, if the contributors into a collaborative software development pursue the goal that their inputs merge into inseparable or interdependent parts of a whole, such collaborative project can qualify as a joint work [12].

Article 2 (2) Software Directive says: "*In respect of a computer program created by a group of natural persons jointly, the exclusive rights shall be owned jointly*" [7]. In principle, and unless agreed otherwise, contributors act as co-owners and enjoy equal rights to license the whole work on a non-exclusive basis subject to accounting obligations [12]. By that, neither contributor holds exclusive rights on his own, but can enforce the copyright [12]. On the other hand, the exploitation of a joint work requires consent of all contributors. If, for instance, one contributor refuses to cooperate with the others and disagrees with the licensing strategy the attempts to exploit such software product stand under the risk of being challenged as copyright infringement [6]. A possible alternative is to rewrite or remove the part of disagreeing party.

In free and open source (FOSS) projects, this issue is settled in a way that all bits are being contributed under the same or compatible license.

C. Derivative Work

Derivative is another type of collaborative work. In general terms, a derivative work builds upon a pre-existing work, creates a new, separate work, includes portions of a prior work and receives an individual copyright [12]. In legal terms, the development of a derivative work on top of a prior work requires authorization of the original right holder to modify his program and develop derivative works. The right to modification belongs to exclusive rights of a right holder, as defined by Article 4 (1) (b) Software Directive: "*the translation, adaptation, arrangement and any other alteration of a computer program and the reproduction of the results thereof, without prejudice to the rights of the person who alters the program;" [7]*.

In contrast to joint ownership, the creator of a derivative work inherits a copyright in it: *"The party executing the new work holds the copyright in the new elements in its own right and a right to control the whole as a unified, copyrightable product." [12]*.

Although, there is an independent copyright in a derivative work, it extends only to the portions derived from the original work and does not affect or prejudice the original copyright. A contributor, who builds on top of a prior work and creates his own derivative, may mark his copyright, for example as follows:

"Copyright © 2018 Project Development Who Made Changes"

The copyright line is normally followed by license notice.

D. Composite Work

A "composite work", also called "collective work" or "compilation", is distinct from the above types. A characteristic element of a composite work is that it is combined from elements that constitute independent and individual works in themselves. An example of a composite work can be a software package combined from a number of programs or modules each separately owned [6]. In contrast to joint works, the parties to a composite work "*do not intend that their contributions be merged to the point of being indistinguishable.*" [12]. Thus, the same program or module can be integrated into different composite works, whereas copyright in such module remains by the contributor.

The exploitation of composite works has some legal peculiarities. In principle, where a software project or package has a number of contributors and/or is made up of a number of individual programs or modules, the exploitation of such product as a whole would normally require consent of all contributors. In other words, if any of the contributors would seek to exploit the product as a unit whereas some contributors would disagree, a disagreeing party or parties may claim copyright infringement. The solutions for handling the situation might be (a) to remove the contribution of the disagreeing party (as Google removed "rangeCheck" from Android); (b) to rewrite a piece of software at issue [6]; or (c) to advance and settle potential IP issues by an agreement.

IV. DATA RIGHTS

Another issue closely associated with the development of data driven software concerns the data rights. This issue is particularly relevant when the data used to train the software is personal health data. The privacy considerations behind the data sharing for medical research deserve a profound elaboration elsewhere and go beyond the scope of this paper [16]. At the same time, the proprietary issues behind the data sharing are relevant for the management of intellectual property rights (IPR) and merit a deeper look here.

As an example of data driven software a Salivary Gland Ultrasonography image segmentation, developed in the HarmonicSS project may be used. The tool is designed for automatic ultrasonography image segmentation for the identification of large salivary glands. Salivary gland ultrasonography (SGUS) is considered as a valuable tool for the assessment of major salivary gland involvement in primary Sjögren's syndrome. The tool will operate against image processing techniques for automatic intensity and texture features extraction for segmenting the large salivary glands. The techniques will be validated by comparing the automatically segmented large salivary glands with those manually annotated by the experts. Usability will be assessed with Software Usability Scale (SUS) and Technology Acceptance Model (TAM) [3]. In the development process, the tool is supposed to be trained, and further validated against the real patient data.

Following the Commission Recommendation on the management of intellectual property in knowledge transfer activities and Code of Practice for universities and other public research organisations [17], the data, which the clinical partner institutions process before starting the project and agree to share for the project research, qualifies as background. The clinical data providers are supposed to hold all necessary rights in the clinical data they contribute. By contributing the data to the project, the data providers also agree (and shall have the legal capacity) to grant access rights to such data as technical partners may request for implementation of the project. And this is the mechanism, how the clinical data enters into the project and may be used for research.

Against this background, the developers of a SGUS image segmentation tool shall seek the access rights to the clinical data they need to train the tool and, if granted, may use the clinical data under the use rights. At a stage when the SGUS tool is developed, the question of copyright ownership arises. An associated issue is how copyright ownership interrelates with the rights in clinical data. In this respect,

following the Commission Recommendation, the ownership of results shall stay with the party that has generated it [17]. In application to the SGUS tool, it means, the software copyright in the SGUS tool shall stay with the developers. As regards the correlation of copyrights with the data rights, the Commission Recommendation provides: "*The ownership of background should not be affected by the project*". What follows is that the rights which data providers hold in clinical data stay by the data providers and are not affected by the copyright, which software developers inherit in the SGUS tool.

V. CONTRACTUAL MANAGEMENT OF IP RIGHTS

Last, but not least, the European Commission when guiding research projects funded by the Commission calls for the management of Intellectual Property (IP) rights preferably at the outset of the project. Accordingly, the Principles regarding collaborative and contract research, established by the European Commission [17], provide "*IP-related issues should be clarified at management level and as early as possible in the research project, ideally before it starts. IP-related issues include allocation of the ownership of intellectual property which is generated in the framework of the project*…" Although, a general rule is that results generated in a collaborative research project should stay with the party that produced the results, the ownership "*can be allocated to the different parties on the basis of a contractual agreement, adequately reflecting the parties' respective interests, tasks and financial or other contributions to the project"* [17].

This approach, namely management of IP rights by an agreement has been adopted by the research project HarmonicSS. The management of IP rights in the project is specifically addressed by an IPR agreement. IPR agreement lays down principles of research, regulates the allocation of rights in data and research results and governs the issue of composite ownership in combined works. The matters of individual and joint ownership are already covered by the contractual framework of the project.

In particular, the IPR agreement defines the concept of composite work, allocates the ownership to contributing partners according to the contribution of each and binds the parties who contribute into composite works and wish to exploit composite works as a whole to agree on the ownership shares, allocation and exercise of rights, sharing of revenues, protection measures and the division of related cost in advance. In the same vein, such issues as the terms of licensing software from collaborative development shall be addressed by the agreement. For instance, an option of dual licensing may be considered, such as: licensing "open source" for research, and proprietary licensing into commercial exploitation. The variable licensing schema is followed by many commercial software providers. One example is Microsoft, offering open programs and commercial licensing agreements [18].

In summary, integral licensing is important for any collaborative software development project, since licensing is the key to successful software exploitation and bringing software right to the right sectors of the market. Such integral licensing may be reached by an agreement between the project participants deciding to license project outcomes under the one licensing schema.

VI. CONCLUSION

This paper investigates the issues of copyright and data ownership in collaborative software projects in application to data driven software. The outcome is the result of legal research conducted in ICT research projects with focus on technical matter but is not technical in itself. The conclusions made in the course of this study follow:

1) Ownership of copyright: Essentially, the legal relations surrounding the creation of a software product are define the ownership of copyright. A developer is a copyright holder if he wrote a program in his capacity as a natural person, for instance, contributed into an open source project in his spare time. By contrast, if a developer contributes a code into a collaborative research project on behalf of the partner institution, the partner institution acts as copyright owner, unless agreed otherwise.

2) Sharing of copyrights: As a rule, in a collaborative software development with multiple contributions, contributors share copyrights. However, the exploitation of collaborative works depends on the type of contributions, and underlying terms. In principle, exploitation of composite works requires consensus of all contributors and is normally managed by an agreement.

3) Data rights: The rights is data, which the clinical data providers agree to share to the project, stay by the data providers. The data rights are not affected by the results generated by processing the data. The rights in data driven results, such as data driven software modules, pass to the developing parties. The rights in the results are without prejudice to the data rights.

4) Management of IP rights: The exclusive economic rights in software are alienable by nature and can be regulated by contractual schemes. The allocation of copyright shares, the exercise of rights among the contributors, the licensing strategy and division of revenues, if applicable, can be governed by an agreement laying down the terms, under which participants agree to contribute.

ACKNOWLEDGMENT

This project has received funding from the European's Union Horizon 2020 Research and Innovation Programme under Grant Agreement No 731944 and from the the Swiss State Secretariat for Education, Research and Innovation SERI under grant agreement 16.0210.

REFERENCES

[1] I. Lishchuk, "Open Source Software and Some Licensing Implications to Consider", International Journal On Advances in Systems and Measurements, v 9 n 3&4, pp. 266-275, 2016.

[2] D. Longo, and J. Drazen, "Data Sharing", Editorials, NEJM 374;3; 21 June 2016.

[3] HarmonicSS, the Project. [online]. [Accessed: 8 June 2018]. Available from: http://harmonicss.eu/.

[4] PONTE, Efficient Patient Recruitment for Innovative Clinical Trials of Existing Drugs to other Indications, Project ID: 247945, funded under: FP7-ICT.

[5] Apache License, Version 2.0. [online]. [Accessed: 8 June 2018]. Available from: http://opensource.org/licenses/Apache-2.0.

[6] C. Reed, and J. Angel, Computer Law: The Law and Regulation of Computer Technology, 6. Edition, Oxford University Press, 2007.

[7] Directive 2009/24/EC of 23 April 2009 on the legal protection of computer programs, OJEU, L 111/16, 5.5.2009.

[8] Berne Convention for the Protection of Literary and Artistic Works, adopted in 1886.

[9] U.S. District Court for the Northern District of California, Ruling of 31 May 2012, Case C 10-03561 WHA, Oracle America, Inc., v. Google Inc.

[10] U.S. Court of Appeals for the Federal Circuit, Ruling of 09 May 2014, Oracle America, Inc., v. Google Inc., Appeals from the United States District Court for the Northern District of California in No. 10-CV-3561.

[11] J. Mullin, "Google beats Oracle—Android makes "fair use" of Java APIs," Ars Technica, 26 May 2016 [online]. [Accessed: 8 June 2018]. Available from: http://arstechnica.com/tech-policy/2016/05/google-wins-trial-against-oracle-as-jury-finds-android-is-fair-use/.

[12] R. Nimmer, Legal Issues in Open Source and Free Software Distribution, adapted from Chapter 11 in Raymond T. Nimmer, The Law of Computer Technology, 1997, 2005 Supp.

[13] D. McGowan, Legal Implications of Open-Source Software, 2001 U. Ill. L. Rev. 241, 268, 274 (2001); Greg R. Vetter, The Collaborative Integrity of Open Source Software, 2004 Utah L. Rev. 563.

[14] UK, Copyright, Designs and Patents Act 1988.

[15] Software Freedom Law Center, Maintaining Permissive-Licensed Files in a GPL-Licensed Project: Guidelines for Developers, 2007. [online]. [Accessed: 8 June 2018]. Available from: www.softwarefreedom.org.

[16] M. Stauch, The Draft Data Protection Regulation and the Secondary Use of Patient Data for Research: Projects amd Concerns. Journal of professional Negligence, 29:72, 2013.

[17] Commission Recommendation on the management of intellectual property in knowledge transfer activities and Code of Practice for universities and other public research organisations, 10.4.2008, C(2008)1329.

[18] Microsoft, Licensing programs. [online]. [Accessed: 8 June 2018]. Available from: https://www.microsoft.com/en-us/Licensing/licensing-programs/licensing-programs.aspx.

 ISBN: 978-1-61208-655-2