INTERNET OF THINGS, INTEROPERABILITY AND INTERFACES: A COPYRIGHT LAW PERSPECTIVE

Matěj Myška

Senior Assistant Professor, Masaryk University, Faculty of Law, Institute of Law and Technology
Veveří 70, 61180 Brno, CZ
matej.myska@law.muni.cz; http://cyber.law.muni.cz

The publication of this paper is supported by the Czech Scientific Foundation – project ID no. GA17-22474S – «Adapting Exceptions and Limitations to Copyright, Neighbouring Rights and Sui Generis Database Rights to Digital Network Environment»

Schlagworte: ***interoperability, software, copyright, decompilation, reverse engineering***

Abstract: *The interoperability of the elements involved in IoT is the factual conditio sine qua non. Practically, the interoperability is ensured via various interfaces. If the specification of an interface is not readily available, it might be under certain strict conditions obtained by reverse engineering. The current EU copyright law however lays down strict prerequisites on the reverse engineering without authorization of the rightholder. In the EU copyright law context, the copyright protection for interfaces is discussed (section 2). Next, the current regulation in computer program directive of reverse engineering of interfaces is presented (section 3). Finally, this regulation is critically evaluated and recommendations how to improve the current situation are offered (section 4). The last part (section 5) concludes.*

1. Introduction and scope

Recent study of the McKinzey Global Institute [Manyika et al. 2015, 23] claims that «*[o]n average, as 40 percent of the total value that can be unlocked requires different IoT systems to work together».* In order to unlock this value and realize the basic functionality of IoT, the involved physical objects need to «talk» to each other – i.e. «*share information and coordinate decisions»* [Al-Fuqaha et al. 2015, 2347]. In other words, the full potential of IoT is only realized when «*diverse elements comprising IoT (devices, communication, services, applications, etc.) […] seamlessly cooperate and communicate with each other»* [Noura et al. 2018, 3], i.e. are interoperable.

The term «interoperability» itself is defined in various ways,[1] but in the IoT the most pertinent definition seems to be the «*ability of two systems to communicate and share services with each other»* [Kiljander et al. 2014, 856]. Specifically, in the IoT, interoperability encompasses various layers [Noura et al. 2018, 3]. Normatively, interoperability, i.e. functional logical and physical interconnection and interaction on logical and physical level, is regarded as the basic function of computer program (recital 10 CPD).[2] Interoperability is ensured via «interfaces». These special parts of computer programs could be again understood in various ways, but this paper discusses the model employed by van Rooijen [2010, 13–16].[3] Accordingly, he distinguishes four types of interfaces: user interfaces, data interfaces, communications interfaces (protocols) and Application Programming Interfaces (APIs) [van Rooijen 2010, 14].

---

1 See Noura et al. [2018] for extended discussion of the issue of interoperability in the context of IoT.

2 Directive 2009/24/EC of the European Parliament and of the Council of 23 April 2009 on the legal protection of computer programs (Codified version) – further referred to as «CPD».

3 Van Rooijen [2010] refers to the now already withdrawn standard for Posix Open Systems Reference Model (ISO/IEC TR 14252:1996 Information technology – Guide to the POSIX Open System Environment (OSE)) presented by Severance [1999]. Weston [2017, 78] emphasizes the «exchange of information» criterion as the defining feature of interfaces and distinguishes the following forms of interfaces: «*application programming interfaces (APIs), protocols, and data file formats».*

As a rule (Art. 1(1) CPD) the expression of a computer program is protected by copyright as literary work, however the ideas and principles thereof, including those underlying interfaces, are not (Art. 1(2) CPD). Due to the indispensability of the interfaces for proper functioning of the computer programs (and IoT in general), as well as functioning competition, the regulation of interfaces and consequently interoperability is a compromise «*between the control by rightsholders and openness of interfaces»* [Weston 2017, 427].

This paper focuses on copyright law as an ex-ante tool how to achieve the desired balance. Specifically, the copyright protection for all of the abovementioned interfaces is discussed (section 2). Next, the current regulation in CPD of reverse engineering of interfaces is presented (section 3). Finally, this regulation is critically evaluated and recommendations how to improve the current situation are offered (section 4). The last part concludes.

This paper does not, however, discuss in detail the history, basic aspects and fundamental notions of the issue at hand, as this has already been done elsewhere. Specifically, no attention is paid to the question of defining IoT, the role of interoperability therein and technical issues.[4] Furthermore, out of the pertinent legal issues, the paper does not discuss in detail the ones connected competition law,[5] that is generally regarded as an ex-post remedy and second relief in achieving interoperability in the context of dominant and this position abusing entity [Weston 2012, 427]. Finally, this paper focuses solely on the legal situation in the European Union, despite the fact that the copyright law issues related to APIs are currently the focal point of the US legal practice and doctrine.[6]

2. Interfaces & copyright law

The CPD is extraordinarily scarce as regards to the basic definitions of protected subject matter – one of the stated reasons being the alleged resistance to becoming outdated [Janssens 2014, 93]. Consequently, there is neither a definition of a computer program, nor of an interface in the CPD. Moreover, the CPD does not differentiate among the different types of interfaces, which is a must, as the further presented conclusions relevant to one type of interfaces are not directly applicable for different types of interfaces [Gervais/Derclaye 2012, 568]. As regards to the negative subject matter, i.e. what is not protected, the CPD does not provide much clarity either – the interfaces are not *expressis verbis* excluded from protection. However, as noted by Janssens [2014, 98] the observable legislative policy is to exclude them on the basis of the «expression/idea dichotomy» principle. The restricted acts (Art. 4 CPD) are namely reserved for «expression» of the computer program in any form, be it a source code or an object code.[7] According to recital 11 CPD «logic, algorithms and programming languages» comprise «principles and ideas», even those underlying interfaces, and are thus not protected under CPD. The provisions of the CPD must be however interpreted in the context of Art. 9(2) TRIPS and Art. 2 WCT and thus also «procedures, methods of operation or mathematical concepts as such» are excluded from copyright protection. The Court of Justice of the European Union (further referred to as «CJEU») addressed the issues of interface copyright protection in merely two cases.

As regards **graphic user interfaces**, in *Bezpečnostní softwarová asociace*[8] the CJEU specifically held, that the graphic user interface does not constitute an expression of a computer program and thus does not enjoy

4 See e.g. Kiljander et al. [2014], Al-Fuqaha et al. [2015], Noura et al. [2018].

5 For detailed discussion of these issues see in detail see e.g. van Rooijen [2010], Wiebe [2011], Kerber/Schweitzer [2017].

6 The «new wave» [von Lohmann 2018] of cases dealing with software interfaces has been initiated by the *Oracle v. Google* saga. The details of this legal proceedings are extensively discussed by Menell [2018]. Samuelson/Scotchmer [2002] offer a historical and economic perspective on these issues in the USA.

7 However, as will be discussed later, the principles and ideas might be expresssed in the code.

8 Judgment of the Court (Third Chamber) of 22 December 2010, Bezpečnostní softwarová asociace – Svaz softwarové ochrany v Ministerstvo kultury, C-393/09, ECLI:EU:C:2010:816, para. 42.

protection granted by the CPD.[9] The arguments for such conclusion included *inter alia* the fact, that such an interface does not enable the user to reproduce the protected computer program.[10] On the other hand, such an interface can be, if it fulfils the criterion of originality, protected by the «*ordinary law of copyright*»[11] stipulated in the ISD.[12] The CJEU also tried to «clumsily» [Gervais/Derclaye 2012, 568] adopt the merger doctrine. Namely, when the expression of the components at hand is dictated by their technical function, the methods of implementation of the idea in the expression are so limited that the idea and expression becomes «indissociable».[13]

Concerning **data interfaces**, in *SAS Institute*, the CJEU held that data formats *«used in a computer program in order to exploit certain of its functions»* (together with programming languages and functionality of the computer program) do not constitute an expression of that program and thus do not enjoy protection granted by CPD.[14] Nevertheless, as in the cases of graphic user interfaces, the data formats may enjoy the «traditional» copyright protection offered by ISD, if they are original, in the sense that they are author's own intellectual creation.[15] To achieve this level of originality would be however rather problematic, as the data formats are by their nature functional and thus not original [Gervais/Derclaye 2012, 569]. These authors [2012, 569] further observe, that the CJEU wanted to say, that the *actual code* underlying the data formats expressing the ideas and principles might be protected (even by CPD).[16] Yet again, the originality threshold in this case would be hard to pass, given the limited ways how to express the data formats. As noted by Weston [2017, 81] this means that rewriting the interface specification without copying the «expressive code» is not to be considered as CPD-granted copyright infringement.

As of 2019 **communications interfaces (protocols)** and **APIs** were not the subject of any decision of the CJEU. It might be reasonably expected, that the CJEU would opt again for the same strategy and declare the underlying principles and ideas in communications interfaces (protocols) and APIs as not an expression of the computer program and/or functional and thus non-protectable under CDP. At the same time the actual code would expressing them would remain protectable subject matter, provided that they are original [Weston 2017, 97; Janssens 2014, 98]. The CJEU might also hold, as in the *Bezpečnostní softwarová asociace* and *SAS Institute*, that they might constitute a «work» under ISD. Especially in the case of APIs, the CJEU however would need to address the most pertinent issue that it is up until now evading, namely the clear separation of specification and implementation of the interfaces and their legal treatment. It is remarkable, that the CJEU has not actually dealt (or at least in a clear and understandable way)[17] with this basic conceptual feature of the interfaces before.

The legal doctrine has on the other hand eloquently grasped the difference between the unprotectable specification of the interface (i.e. the *«rules and method of interaction»)* and protectable implementation thereof (i.e. actual the «*implementation of the interface specifications into a program's code*») [Palmer/Vinje 1992, 69].[18] Consequently, the specification could be implemented independently by a different programmer [Weston 2012, 435–436] and obtain copyright protection for this new original expression. However, a prerequisite is

---

9 However, the CJEU did not specifically address the question, whether the underlying code creating the interface is protected [Weston 2012, 439].

10 *Bezpečnostní softwarová asociace*, para. 41.

11 *Bezpečnostní softwarová asociace*, para. 44.

12 Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society – further referred to as «ISD».

13 *Bezpečnostní softwarová asociace*, para. 49.

14 Judgment of the Court (Grand Chamber), 2 May 2012, SAS Institute Inc. v World Programming Ltd, C-406/10, ECLI:EU:C:2012:259, para. 39.

15 *SAS Institute*, para. 45 – referring to *Bezpečnostní softwarová asociace*, paras. 44 to 46.

16 See also *SAS Institute*, para. 43.

17 The *SAS Institute*, para. 43 might be understood as a rather cumbersome effort to delimit these issues.

18 See further e.g. the conclusions presented by Weston [2017, 97]; Janssens [2014, 98].

that there are (i.e. the idea and expression do not merge as noted by the CJEU in *Bezpečnostní softwarová asociace*).

3. Reverse engineering for interoperability and copyright law

In copyright law, the abovementioned «openness of interfaces» is realized in two-fold way – firstly, by limiting the scope of protection as discussed above (section 2) and secondly by specifically excepting various restricted acts from the authorization of the rightholder (Art. 5 and 6 CPD). The rationale behind this regulation is the need to find and appropriate ex-ante remedy,[19] when the required interface specification is missing. In a standardly distributed computer program (i.e. in the form of compiled and executable code) this is the rule – the required interface specifications are namely embedded into the object code [van Rooijen 2010, 68] and unreadable for a human.

This form of code could be however reverse engineered (or reverse analysed) from this object code in order to obtain the needed interface specifications.[20] The CPD contains both of the applied forms, i.e. «black box» reverse engineering (Art. 5(3) CPD) and «white box» reverse engineering, also called «decompilation» (Art. 6 CPD). Both of these procedures entail otherwise restricted act of reproduction (both permanent as well as temporary). Furthermore, both the Art. 5 CPD as well as Art. 6 CPD do not stipulate an active obligation of the rightholder to disclose the information regarding interfaces, but merely stipulates a «duty to tolerate» [Kerber/Schweitzer 2017, 57]. Moreover, according to the Art. 8(1) CPD both the «black box» and «white box» reverse engineering is immune to contractual override.

The «**black box» reverse engineering** procedure covers all reverse analysis techniques short of decompilation [Palmer/Vinje 1992, 78] and comprises of «extensive observation of the ‹box›», i.e. the computer program [van Rooijen 2010, 68]. According to the Art 5(3), the person having a right to use a copy thereof is entitled to *«observe, study or test»* the functioning of the «box» in order to find out, how it works, that is to discover the underlying ideas and principles. These activities must be realised however only during the allowed acts of loading, displaying, running, transmitting or storing the program (Art. 5(3 CPD)). All of these acts necessarily create at least a temporary copy, that is however sanctioned by the CPD. As opposed to the *infra* discussed decompilation, this exception however does not justify access to source code [Bently/Yin-Harn 2016, 261]. The interface specification could be thus obtained by observing and studying the input/output of the computer program. Contrary to the Art 6 CPD, reverse engineer who relies on Art. 5(3) CPD is not restricted in the way, how to handle the information obtained [Weston 2012, FN 123].

The **decompilation exception** (Art. 6 CPD)[21] was arguably one of the most controversial, lobbied and contested ones in the CPD – the reason for it being the alleged supporting of software piracy.[22] It allows the person having a right to use a copy of a program «*to look at and understand the basic building blocks of the program*» [Weston 2012, 427] for the sole purpose of achieving interoperability with other program.[23] This «looking at & understanding» also comprises acts, that would be otherwise restricted – reproduction and translation of the code – and that are under specific requirements set in the Art. 6 CPD deemed as non-infringing. It must be emphasized, that Art. 6 CPD refers only to «code», not computer program as such. As a result, i.e. preparatory design materials must not be copied and/or translated. The core regulatory principles of the Art. 6 CPD comprise the indispensability criterion and the proportionality criterion [Kerber/Schweitzer 2017,

[19] Further options how to obtain the needed interoperability information include voluntary disclosure thereof and forced disclosure via the remedies provided by competition law.

[20] The actual process itself is however costly and lengthy – if the potential infringer wanted to create illicit copies of the computer program it is less expensive to write it from scratch than to try to copy it via reverse engineering [Johnson-Laird 1992, 348; similarly Wiebe 2011, 95].

[21] Bing refers to «decompilation» as right [2009, 17].

[22] For and excellent overview of the legislative history thereof see Palmer/Vinje [1992] and Band [2018].

[23] I.e. not to data interfaces or hardware [van Rooijen 2010, 88].

56]. The decompilation is indispensable, when the necessary information is not available via other means, including through «black box» analysis (Art. 5(3) CPD) [Blocher/Walter 2010, 172]. The decompilation is thus the «last resort» to which the creators of other programs wanting to achieve interoperability should turn. From a technical point of view engineering technically necessary, whenever the information is incomplete or inaccurate [Johnson-Laird 1992, 345]. As a result, the rightholders might effectively «phase out» this provision by providing complete specification of the interface. Such behaviour of the rightholders could be regarded as *status idealis*. Otherwise, the burden of proof regarding the availability of the interoperability information rests with the decomplier [Weston 2017, 102; van Rooijen 2010, 90–91].[24] Moreover, according to the Art. 6(1)(c) CPD the decompilation should be limited only to the parts of the original program, that are necessary in order to achieve interoperability. The legal doctrine [van Rooijen 2010, 91; Bently/Yin-Harn 2016, 261] rightly pinpoints the logical fallacy of this requirement – standardly the decompiling engineering does not know, where are the specifications of the interfaces are located in the object code. The main reason, why the decompilation is undertaken is exactly to find this out. *Lege artis* the decompiling person should thus firstly employ the «black box» analysis first to «*approximately locate the location of the relevant interfaces*» [van Rooijen 2010, 91] and only afterwards employ decompilation. If this procedure does not turn out to be successful, the condition should not apply [van Rooijen 2010, 91]. The decompiling person is however restricted in dissemination of the lawfully obtained information (sic!) (Art. 6(2)(b) CPD). Generally, it shall not be used for any other purposes than achieving interoperability. Next, it shall not be given to others, «*except when necessary for the interoperability of the independently created computer program*» (Art. 6(2)(b) CPD). And finally, the obtained information must not be used for creation and marketing of substantially similar computer program. The proportionality criterion is expressed in the Art. 6(3) CPD that subjects the interpretation of the decompilation provision to the three-step test (i.e. certain special cases – no conflict normal exploitation – no prejudice of the legitimate interests of the rightholder).

4. Criticism, discussion and suggestions de lege ferenda

Since its inception, the regulation of reverse engineering for interoperability is not without criticism. The consequently the regulation as such is deemed to be overprotective [Weston 2017, 84] and too restrictive (or too narrowly construed [Wiebe 2011, 95]) and as a result significantly limiting the abilities of the competitors to access the needed interoperability information [van Rooijen 2010, 91].

van Rooijen [2010, 69–70] offers the concise critique and explains the overprotection phenomenon as regards to interfaces. In his view the root of the problem lies in three specifics of the computer program's copyright protection as literary work. Firstly, the current regulation protects also the closed code expression of the computer program – thus the mere *access* to the work is controlled by the rightholder [van Rooijen 2010, 69–70]. Secondly, the reproduction right is broadly defined and consequently enables the conditioning of reverse engineering [van Rooijen 2010, 69–70]. Thirdly no positive obligation of the rightholder to provide access to his work is stipulated [van Rooijen 2010, 69–70]. Combined, the CPD shows the inadequacy of copyright protection for computer programs. By (over)protection of the shell (or tissue), the regulation is actually trying to protect, what is the most valuable on computer programs, namely the underlying know-how (even ideas and principles) and functionality, which is otherwise out of scope of copyright protection.[25]

In the context of IoT the most obvious disadvantage of the current regulation is the limitation of reverse engineering only for achieving «*interoperability of an independently created computer program with other programs*» (Art 6(1) CPD). Reverse engineering for production of interoperable hardware is expressly excluded [Blocher/Walter 2010, 180].

[24] Palmer/Vinje [1992, 81] however claim that this issue is left to the Member State's national implementation of the CPD.

[25] As was also confirmed by the CJEU in *SAS Institute.*

Another subject of criticism is the regulation of dissemination of the information obtained by decompilation (Art. 6(2)(b) CPD). This provision is claimed to be prohibitively restrictive and creating a «statutory trade secret law» limiting the sharing of the interface specification [Weston 2017, 82]. Moreover, it is regarded as unique regulation in the copyright law system that protects and prevents use and dissemination information and not an original work [Bently/Yin-Harn 2016, 261].

Finally, the general overarching condition of the compulsory interpretation compliant with the three-step is (Art. 6(3) CPD) yet again is too restrictive. As argued by legal doctrine ([Kerber/Schweitzer 2017, 57] referring to [Wiebe 2011, 92]) the allowed reverse engineering does not hamper rights holder's legitimate interest in retaining a competitive lead – it is protected by the simple technical complexity.[26]

An ideal solution within the system of copyright law to the identified problems would be thus the changing of «hard law», i.e. the CPD. Ex-ante explicit regulation on the scope of protection for both specifications as well as implementation of interfaces is advisable [Weston 2012, 447]. As a result, the IoT market entrants would not have to rely on the ex-post competition law remedies (Art. 102 TFEU) against the abusing dominant competitor. Further advisable changes might involve the mandatory disclosure of interface specification by the rightholder, i.e. to grant access to interface specification. Yet another suggested change includes the easing of restrictions applicable to dissemination of the reverse engineered information [Weston 2017, 105].[27] A further step might be the exclusion of implementation code (i.e. also the expression, not only the underlying principles and ideas) from copyright protection [Weston 2017, 97–98]. A fundamental change is suggested by Wiebe [2011, 93], namely the complete exclusion of reverse engineering activities from the scope of copyright law protection – consequently, this activity should not be restricted to achieving of interoperability but should be allowed also for the purposes of education or research, maintenance as well as security purposes. This would also address the most problematic limitation, i.e. the program-to-program interoperability requirement. This last issue should be however solved even without such fundamental changes in the IoT context, reverse engineering for data interfaces/hardware interoperability is most advisable.

However, given its legislative history and the related controversy with its adoption, the rather radical change of the CPD is highly unlikely [Weston 2017, 99]. In these circumstances, the «soft law» approach is to be regarded as more feasible way, how to promote interoperability. Such policy change should encourage the culture of sharing interfaces [Weston 2017, passim]. This could be achieved by propagating the use of open specification standards.[28] Indirectly, this shift might be encouraged by advertising the availability of the interface specifications e.g. by creating a single EU contact point [Commission Staff Working Document 2013, 15]. Finally, the sharing might be also incentivized by subjecting the public funding related to computer programs[29] to availability of such information.

5. Conclusions

Kerber/Schweitzer [2017, 40] note, that the expected benefits of IoT and Industry 4.0 «*hinge on the interoperability between networks, software and data*». The study of the McKinzey Global Institute mentioned in

[26] The technical complexity logically results into high financial and time costs. Furthermore, as aptly noted by van Rooijen [2010, 87] the reverse engineering process itself is uncertain of results.

[27] In this regard Blocher/Walter [2010, 175–179] offer an interesting interpretative solution of the alleged problem of the Art. 6(2) CPD. They understand the term «information» in this specific paragraph as only «information» protected by copyright [Blocher/Walter 2010, 176]. Protection of information – i.e. a concept alien to copyright – cannot be inferred from the lack of permission in dealing with it [Blocher/Walter 2010, 176]. Furthermore, they present the Art. 6(2) CPD as actually constituting free uses for the decompiling engineer as regards to the copyright protected «*information and code parts*» [Blocher/Walter 2010, 177]. Consequently, she is free to a) use it to achieve interoperability of the other independently created program; b) give it to other when necessary to achieve interoperability with such program and c) use it for creation and marketing of substantially similar computer program [Blocher/Walter 2010, 177].

[28] See Li [2018] for in-depth discussion of the issue of standards and open standards and relevant copyright law issues.

[29] Both procured as well as created within the publicly funded research grants.

the introduction even claims that «*interoperability is critical to maximizing the value of the Internet of Things*» [Manyika et al. 2015, 23]. Furthermore, lack of interoperability increases costs for the potential competitors and effectively reduces the needed competition [Weston 2012, 428].

The legal regulation of interoperability is a balancing exercise between control granted by copyright and the needed pro-competition openness of interfaces [Weston 2017, 89]. This short paper tried to show that the current EU copyright framework currently favours the control and does not follow suit the perceived importance of interoperability in IoT. In sum, the reverse engineering provisions are extremely complicated and do not offer much clarity to the already technically very demanding activity. The CJEU also did not provide for much clarity with its rather evasive approach. As a result, the current regulation does not provide the market entrants, trying to develop interoperable IoT solutions, with the needed legal certainty.

Firstly, the regulation does not contain clear rules on the scope of protection itself, especially as regards to APIs. As was shown in the section 2 of this paper, the underlying ideas and principles, but also procedures, methods of operation or mathematical concepts as such shall not be protected by copyright law, however their original expression in the code might as well be. Due to their functional nature, the copyright protection for interfaces however seems to be rather an exception. Either, these would not be expressed in code as a computer program (as might be the case in data interfaces and communication interfaces); or, in the case of APIs, the idea (interface specification) would probably merge with is expression (interface implementation).

Secondly, the conditions set for legal reverse engineering (section 3) are too restrictive in comparison to the actual practice and effect of reverse engineering (section 4). In the context of IoT the prohibition of «white box» reverse engineering for the program-to-hardware and data-to-hardware interoperability seems to be crucial. Furthermore, the prohibition of dissemination of the already obtained information is economically unsound and ineffective and thus not fostering competition [Weston 2017, 100–101].

The section 4 of this paper thus sketched some suggestions, how to overcome the identified problems. These might include amending the CPD as well as «soft» measures encouraging the sharing of interoperability information.

The pragmatic assessment of current decompilation provision (Art. 6 CPD) presented by Bing [2009, 424] seems to be still valid. He – also due to the considerable difficulty and resource intensity of this activity – claims, that the major effect thereof is allegedly that on legal policy – namely, it actually incentivizes the rightholders to «*publication of interface specification and through this, exclude the application of the provision*». Otherwise than that, the current CPD rather overprotects interfaces and stifles innovation.

6. References

Al-Fuqaha, Ala/Guizani, Mohsen/Mohammadi, Mehdi/Aledhari, Mohammed/Ayyash, Moussa, Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications, IEEE Communications Surveys Tutorials, volume 17, issue 4, 2015, pp. 2347–2376. DOI:10.1109/COMST.2015.2444095.

Band, Jonathan, The Global API conflict, Harvard Journal of Law & Technology, volume 31, Special Issue Spring 2018, pp. 615–637.

Bently, Lionel/Yin-Harn Lee, Directive 2009/24/EC – on the legal protection of computer programs (Computer Programs Directive). In: Dreier, Thomas/Hugenholtz, P. Bernt (Eds.), Concise European Copyright Law, Second edition, Alphen aan den Rijn Kluwer Law International, 2016, pp. 421–490.

Bing, Jon. Copyright protection of computer programs. In: Derclaye, Estelle, Derclaye, E. (ed.), Research Handbook on the Future of EU Copyright, Edward Elgar Publishing, Cheltenham/Northampton 2009, pp. 401–426.

Blocher, Walter/Michel M. Walter, Computer Program Directive, In: Walter, Michel M./Lewinski, Silke von (eds.), European Copyright Law: A Commentary, Oxford University Press, Oxford 2010, pp. 81–248.

Commission Staff Working Document, Analysis of measures that could lead significant market players in the ICT sector to license interoperability information, Brussels, 6. 6. 2013, SWD(2013)209 fin.

Gervais, Daniel/Derclaye, Estelle, European Intellectual Property Review, The scope of computer program protection after SAS: are we closer to answers?, volume 34, issue 8, pp. 565–572.

Janssens, Marie-Christie, The Software Directive. In: Stamatoudi, Irini A./Torremans, Paul, EU Copyright Law, Edward Elgar Publishing 2014, pp. 89–148.

Johnson-Laird, Andrew, Reverse Engineering of Software: Separating Legal Mythology from Actual Technology, Software Law Journal, volume 5, 1992, pp. 331–354.

Kiljander, Jussi/D'Elia Alfredo/Morandi, Francesco/Hyttinen Passi, Takalo-Mattila Janne/Ylisaukko-Oja Arto/Soininen/Juha-Pekka/Cinotti, Tullio Salmon, Semantic interoperability architecture for pervasive computing and internet of things, IEEE Access, volume 2, 2014, pp. 856–873. DOI:10.1109/ACCESS.2014.2347992.

Li, Jingze, Intellectual property licensing tensions: utilising open source software in the formal standard-setting context, European Journal of Law and Technology, volume 9, issue 2, 2018, http://ejlt.org/article/view/593.

Manyika, James/Chui, Michael/Bisson, Peter/Woetzel, Jonathan/Dobbs, Richard/Bughin Jacques/Aharon, Dan. The Internet of Things: Mapping the Value beyond the Hype. https://www.mckinsey.com/~/media/McKinsey/Business%20Functions/McKinsey%20Digital/Our%20Insights/The%20Internet%20of%20Things%20The%20value%20of%20digitizing%20the%20physical%20world/The-Internet-of-things-Mapping-the-value-beyond-the-hype.ashx (accessed on 8 January 2019), McKinsey Global Institute, June 2015.

Menell, Peter S., Rise of the Copyright API Dead?: An Updated Epitaph for Copyright Protection of Network and Functional Features of Computer Software, Harvard Journal of Law & Technology, volume 31, Special Issue Spring 2018, pp. 305–490.

Noura, Mahda/Atiquzzaman, Mohammed/Gaedke, Martin, Interoperability in Internet of Things: Taxonomies and Open Challenges. Mobile Networks and Applications, 2018, pp. 1–14. DOI:10.1007/s11036-018-1089-9.

Palmer, Alan/Vinje, Thomas, The EC Directive on the Legal Protection of Computer Software: New Law Governing Software Development, Duke Journal of Comparative & International, volume 2, issue 1, 1992, pp. 65–88, https://scholarship.law.duke.edu/djcil/vol2/iss1/3.

Samuelson, Pamela/Scotchmer, Suzanne, The Law and Economics of Reverse Engineering, Yale Law Journal, 2002, volume 111, issue 7, pp. 1575–1664.

Severance, Charles r., Posix: a model for future computing, Computer, volume 32, issue 1, pp. 131–132.

van Rooijen, Ashwin, The software interface between copyright and competition law: a legal analysis of interoperability in computer programs, Wolters Kluwer, Austin 2010.

von Lohmann, Fred, The New Wave: Copyright and Software Interfaces in the Wake of Oracle v. Google, Harvard Journal of Law & Technology, volume 31, Special Issue Spring 2018, pp. 517–533.

Weston, Sally, Improving interoperability by encouraging the sharing of interface specifications. Law, Innovation and Technology, volume 9, issue 1, 2017, pp. 78–116. DOI: 10.1080/17579961.2017.1302695.

Weston, Sally, Software Interfaces – «Stuck in the Middle: The Relationship Between the Law and Software Interfaces in Regulating and Encouraging Interoperability», IIC International Review of Intellectual Property and Competition Law, volume 43, issue 4, 2012, 427–450.

Wiebe, Andreas, Interoperabilität von Software: Art. 6 der Computerprogramm-Richtlinie aus heutiger Sicht, Journal of Intellectual Property, Information Technology and Electronic Commerce Law, volume 2, issue 2, 2011, pp. 89–96. URN:nbn:de:0009-29-30812.