

Software as Patentable Subject Matter

Analysis of European and U.S. Jurisdictions

by Tornike Kobakhidze

LLM Capstone Thesis SUPERVISORS: Jessica Charles Lawrence & Tommaso Soave Central European University

> © Central European University 07.06.2020

Contents

Introduction
Chapter 1 Patenting Software in the US: A Focus on the <i>Alice</i> Ruling
1.1 The Alice Framework and Related Case Law
1.1.1 Bilski – Revival of "Abstract Idea" Concept7
1.1.2 Mayo – Introducing the 2-Step Test
1.1.3 Alice Corp. v. CLS Bank International
1.1.3.1 <i>Alice</i> Test Step 1. Are the patent claims directed to a patent-ineligible concept – laws of nature, natural phenomenon, abstract idea?
1.1.3.2 <i>Alice</i> Test Step 2. Does a patent claim involve an inventive concept which makes the abstract idea patent-eligible?
1.1.4 After Alice: A Glimmer of Hope for Software Industry 16
1.1.4.1 Categories of abstract ideas – USPTO Guidance
1.1.4.2 When is the claim not directed to an abstract idea? - <i>Enfish, LLC v. Microsoft Corp.</i> example
1.1.4.3 How do you satisfy step 2 of the <i>Alice</i> test? – <i>DDR Holdings</i> and <i>Bascom</i> examples
1.2 Conclusion
Chapter 2 – Patenting Software in Europe
2.1 Computer Programs as Excluded Subject Matter
2.1.1 Technical Character for Computer-Implemented Inventions
2.1.2 Computer Programs and "Further Technical Effect"
2.1.3 Computer-Implemented Inventions versus Computer Programs
2.1.4 Attempt to Raise the Threshold – Inventive Step and Technical Character
2.2 Conclusion
Chapter 3 Comparison and Implications
Conclusion
Bibliography

Abstract

The significance of the software industry has been growing at an immense speed. Due to its unique nature, world's legislators have faced a tough challenge of regulating this field. This issue has been particularly intensified in the field of intellectual property (IP) protection. Software is a creation of the human mind and thus it qualifies for IP protection. However, the IP offices have found it difficult to catch up with the ever-expanding nature of the software. Eligibility of the software for copyright protection is not disputed, however, its patentability is ambiguous. Legal certainty in this field is of an utter importance because software industry needs a consistent legal environment to flourish. An uncertain legal environment undermines its prosperous economic potential, stifles innovation and prevents the effectiveness of the public debates about policy implications.

There are several lacunas regarding software patentability rules in the existing systems and the focus of this paper will be the analysis of those rules, identification of the issues and determination of a superior jurisdiction based on the perspective of the legal certainty. The laws of the U.S. and Europe will be analyzed, because large part of the software developers is located in those jurisdictions and also the differences between those systems clearly illustrate the importance and the complicated nature of the topic. The main source of analysis will be the case law of the respective jurisdictions, because, considering the speed this industry evolves at, it's difficult for the legislation to catch up. The analysis of the jurisdictions will show that both systems lack the certainty and predictability which endangers the software industry as a whole. The paper will briefly discuss the implications of the current approaches. The comparison will conclude that European jurisdiction is superior to the U.S. one, because it offers more certain legal environment.

Introduction

The recent decades have witnessed an explosive expansion of the software industry, which has had a paramount influence on economic development of numerous countries. Tech-based startups are gaining increasing popularity and their significance grows proportionally. This field is becoming one of the most attractive entry platforms and target for many venture capitalists/private equity firms, as it offers prosperous growth not only on domestic, but on global level. Software industry creates new jobs, improves productivity and overall economic potential. According to the BSA Foundation report, the software industry's total value-added GDP in the U.S. amounts up to \$1.6 trillion and it drives growth in nearly every industry.¹

Software is considered as a creation of the human mind, so it is eligible for intellectual property protection. Software can be anything from applications and systems, to programs. Due to its rapid, dynamic growth and ever-expanding nature, legal systems and intellectual property (IP) offices have found it hard to catch up with its development. Technology always reinvents itself and usually replaces the implementations and ideas on which it is based.² Software not only brings life to these technologies, but also changes its nature and processes. Artificial intelligence (A.I.), machine learning, robotics and various modern tech are continuously requiring updated interpretation of the law. IP offices may face question whether the development of technologies affect the way they conceive IP protection of software. Big data, internet of things (IoT), 3D printing, A.I. are concepts that overturned the idea of what human mind alone can achieve. They

¹ BSA Foundation, 'Software: Growing US Jobs and the GDP' (2018) < <u>https://software.org/reports/software-growing-us-jobs-and-the-gdp/</u>> accessed 8 May 2020.

² Connor M. Clyde, 'Current Issues in Technology and Law' (2015) <<u>https://www.mckercher.ca/blog/current-issues-in-technology-and-the-law</u>> accessed 15 April 2020.

involve issues from numerous fields like IP, privacy laws, data protection, security, etc. Software can be a final product, a process or just an improvement and this could possibly create ambiguity in regard with the traditional understanding of an invention. Huge amount of data is being generated every day and software plays crucial role in it. Due to these, director general of World Intellectual Property Organization (WIPO) suggested that even the redefinition of IP rights may become necessary, as the existing ones may not be reliable anymore because of the immense possibilities offered by software industry.³ Notwithstanding these difficulties faced by the industry, patent and copyright remain the most common means to protect software due to reasonable efficiency and lack of other options.

Software qualifies for copyright protection, but its patentability is ambiguous. There has been debates about the potential disadvantages of patent protection and whether the stringent protection stifles innovation instead of its promotion. This controversy is not going to be the focus of the paper. Whatever position one takes, it is certain that the software industry needs a consistent legal environment to operate efficiently. However, there are several lacunas in existing systems which endanger the industry. They create uncertainty and unpredictability. An uncertain legal environment in the field of software can stifle innovation. Patent applicants may find it difficult to predict the outcome of their applications because of fuzzy judicial tests. The need for consistent legal environment is increasing proportionally along with the growth of the industry itself. Uncertainty can have a chilling effect not only on the development of the industry, but it also undermines public debates about software policy implications. Defining appropriate boundaries of

CEU eTD Collection

³ 'Francis Gurry on the future of intellectual property: opportunities and challenges', WIPO Magazine, 2017.

software patentability is critically important and legal uncertainty harms possibility of effective discussions about this controversy.

This paper will focus on the hurdles a software invention has to take to qualify as patenteligible creation. The scope of the paper will cover U.S. and European jurisdictions because of the following reasons: First, because they have different approaches and standards; Second, because these differences clearly illustrate the complexity of the topic and third, because the large part of the software innovators of the world are located in these jurisdictions. Every time this paper refers to the European approach, it means the law of the European Patent Office (EPO). The aim of the comparison is to find a superior jurisdiction for the new entrants in the market. The superiority will be determined solely from the perspective of the legal certainty and other considerations will not be taken into account.

Both jurisdictions will be discussed in the same manner. The paper will not explain the original patentability requirements at length, but rather, will address the software-specific issues arising from the eligibility perspective. The argumentation will mostly rely on the case law of the U.S. Federal Courts, the U.S. Supreme Court and the Boards of Appeal of the European Patent Office. However, several provisions from the United States Code (U.S.C.) and European Patent Convention (EPC) will be used to introduce the topic. It has to be taken into account that legislation cannot catch up with the rapid development of technologies and thus, case law is more reliable source of research in this field of law.

The U.S. chapter will explain the patentability test used by the courts in software patent claims. Argumentation will rely on several cases which are relevant to understand the test. Key terms will also be addressed by the reasoning of the different decisions of the courts. The European chapter will cover the relevant provisions from EPC and explain the EPO approach in the same

manner. Ambiguous terms will be defined by referring to the several decisions of the Boards of Appeal.

The chapter of comparison will briefly compare the law, consequences and rationale behind different approaches adopted by these jurisdictions. The paper will conclude that the European approach is superior to the U.S. one as it is more certain and predictable.

Chapter 1 Patenting Software in the US: A Focus on the *Alice* Ruling

U.S. patent law defines what may be patented in 35 U.S.C. § 101: "Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title."⁴ Although this definition is broad, historically the U.S. courts have defined three judicial exceptions from this general rule. Those are: natural phenomena, laws of nature and abstract ideas.⁵ As this paper focuses particularly on software, I'm not going to discuss the original patentability requirements and exceptions of U.S.C. § 101. Instead, I will explain the requirements for software to be eligible for patent protection set by the case law. Practice has shown that software falls within the literal wording of "process" in § 101, because it is "series of processes based on formalized rules or on objects".⁶ Courts' approach has been constantly changing and the latest ruling of the Supreme Court, which seems to be the most relevant in present, is *Alice Corp. v. CLS Bank International*⁷ case. Although the court in *Alice* does not focus specifically on software patentability, this case has been decisive for software patent applications. The reasons will become clear in the discussion below. As the patentability test used in *Alice*

CEU eTD Collection

⁴ 35 U.S.C. § 101.

⁵ Abraham Kasdan, 'Can You Patent Software and Business Methods in the U.S. - How Did We Get Here and Where Do We Now Stand' (2014) 24 Fed Cir BJ 649 650.

⁶ American Bar Association, 'A Virtual Analog Rule for Software Patent Eligibility' (*Landslide Magazine*, 2015) <<u>https://www.bradley.com/insights/publications/2015/11/a-virtual-analog-rule-for-software-patent-eligib_</u>> accessed 28 March 2020.

⁷ Alice Corp. v. CLS Bank International [2014] S. Ct. 573.

seem obscure, this chapter will explain the rationale behind it. References will be made to several cases to better understand the test, however, the focus will stick to *Alice*.

1.1 The Alice Framework and Related Case Law

The Supreme Court in *Alice* used a two-step test to determine the patentability of software.⁸ Step one focuses on determining whether patent claim is directed to a patent-ineligible concept – an abstract idea. If it is so, then the analysis proceeds to the 2^{nd} step – determining whether the patent claim involves an inventive concept which transforms the patent-ineligible claim into patent-eligible one. Supreme Court has gone a long way to reach the current approach. However, it did not re-invent everything in *Alice*, rather, it combined the logic and reasoning of some previous cases. For this reason, I will first shortly refer to *Bilski*⁹ and *Mayo* cases, as the Supreme Court's analysis in those cases may clarify the court's reasoning in *Alice*. The following two subsections do not intend to explain the rules, rather, they will show the chronology – a way to the current approach. Any unclear concepts used here will be addressed in *Alice* section.

⁸ It should be noted here that this test was first introduced in *Mayo* case (*Mayo Collaborative Servs. v. Prometheus Labs., Inc.* [2012] 132 S. Ct. 1289). However, in *Mayo*, the issue did not involve determining what was an abstract idea and therefore, that case itself did not make a relevant impact on software industry. *Alice* case, by applying the *Mayo* test and meanwhile adapting it to abstract ideas, affected the software patent applications. The reason is that in software patents, the concern about patentability is related to the concept of abstract ideas. ⁹ *Bilski v. Kappos* [2010] 561 U.S. 593, 609.

1.1.1 Bilski - Revival of "Abstract Idea" Concept

The concept of "abstract idea" is an important clue examined by the *Alice* test. The short brief of the case which stressed its importance before *Alice* is provided in this subsection.

In *Bilski*, the patent claim involved a business method for hedging against the price fluctuation risks in the market. The Federal Circuit applied machine-or-transformation test (step one – determination of whether the process/method is tied to a particular machine or apparatus in a non-conventional manner and step two – whether it transforms article from one state to another) and decided that this method was not patentable. The requirements of the test were not met, because this method was not "tied to particular machine or apparatus" and did not "transform a particular article into different state or thing".¹⁰ In other words, Bilski's method was not tied to an implementation in a specific device and did not transform the matter from one physical state to another, either.¹¹ Machine or transformation test is less important for understanding *Alice* case, however it led to the Supreme Court's subsequent decision, denying the sole-importance of this test and introducing the abstract idea test once again, for the first time in nearly 30 years.¹²

The Supreme Court agreed with the result of Federal Circuit, however, it stated that machine-or-transformation test is merely an "important and useful clue".¹³ The more important fact was that the patent attempted to claim an "abstract idea" - the concept of hedging against the risk of price fluctuations. Justice Stevens, in his concurring opinion, stated that: "even if the machine-or-transformation test may not define the scope of a patentable process, it would be a

¹⁰ In re Bilski [2008] (Fed. Cir.) 545 F.3d 943, 954.

¹¹ Christian R. Ruiz, 'Patentable Software: Analyzing Alice Under a law-and-Economics Framework and Proposing a new Approach to Software Claims', (2017) B.U. J. SCI. & TECH. L, Vol. 23:208, 222.

¹² Robert Daniel Garza, 'Software Patents and Pretrial Dismissal Based on Ineligibility' (2018) 24 Rich JL & Tech 1, 29.

¹³ Bilski (n 9) 8.

grave mistake to assume that anything with a 'useful, concrete and tangible result' may be patented".¹⁴ The Supreme Court gave some clues as to what may constitute to an abstract idea, but refused to provide a precise definition. These clues will be discussed below, in *Alice* case section.

1.1.2 Mayo¹⁵ – Introducing the 2-Step Test

The test used in *Alice* was first introduced in this case. However, the claims here were not directed to an abstract idea, but to the "laws of nature".

The patent in *Mayo* claimed a method which measured metabolites in human blood. It calibrated the dosage of triopurine drugs in the treatment of autoimmune disorders.¹⁶ The Supreme Court stated that the claims here were directed to the "laws of nature" (one of the three judicial exceptions from U.S.C. § 101) and they were just instructions for doctors about how to apply the applicable laws during treating their patients.¹⁷ However, such exceptions are not automatically patent-ineligible – even law of nature can become patent-eligible, if the claim does more than just stating the law of nature and adding the words "apply it by computer" in the claim.¹⁸ In other words, there must be an "inventive concept" which ensures that in practice the patent will amount to significantly more than a patent upon the natural law itself.¹⁹ Court stated that the method described in the patent claim was well-understood and routine and therefore, was not "enough" to make the law of nature patentable.²⁰ Thus, the rule is that to transform claim into patent-eligible

CEU eTD Collection

¹⁴ Bilski (n 9) 2 (Justice Stevens, concurring).

¹⁵ Mayo Collaborative Servs. v. Prometheus Labs. Inc. [2012] S. Ct. 132 1289.

¹⁶ Kevin J. Hickey, 'Patent-Eligible Subject Matter Reform in the 116th Congress' (2019) Congressional Research Service, R45918 17.

¹⁷ *Mayo* (n 15) 10.

¹⁸ Ibid.

¹⁹ Garza (n 12) 13 (citing *Mayo*).

²⁰ Hannes Westermann, 'How to treat software in the intellectual property framework' (Bachelor thesis, Lund university 2016) 17.

invention, one is required to do more than applying the method in a generic way.²¹ Court stated that there were three types of limitations to the unpatentable law of nature here: 1) The method was limited to a particular audience, namely, the doctors; 2) Giving the doctors suggestions which laws should be taken into account when treating the patient and 3) adding a conventional, obvious pre-solution activity.²² Obviously, none of these were enough to transform the claim into patent-eligible one.

Mayo case was not about software patents, but it played a significant role in *Alice*. The Court in the latter used the reasoning of *Mayo* and construed the 2-step test which represents the current approach.

1.1.3 Alice Corp. v. CLS Bank International

The *Alice* case is the most important precedent which determines the framework of software patentability in the U.S. The Supreme Court in this case raised the patentability threshold very high, however, it did not completely exclude software from patent protection.²³

The facts of the *Alice* case were as follows: CLS Bank International brought a declaratory judgment action against Alice Corp. It sought to invalidate several patents held by the latter. "The patents disclosed the computerized means of mitigating the risk that one party to a financial transaction may not have sufficient funds to complete the transaction".²⁴ The claimed software

²¹ Kasdan (n 5) 664.

²² Bernard Chao, 'Finding the point of Novelty in Software patents' (2013) Berkeley Technology Law Journal, , Vol. 28:1217, 1234.

²³ Tina Mazalin, 'A comparative analysis of the patentability of software in the U.S. and Europe' (Master thesis, Leibniz Universität Hannover 2017) 41.

²⁴ Michael Macagnone, 'Fed. Cir. Wrestles with Abstraction in Network Patent Case' (*LAW360*, 2015) <<u>http://www.law360.com/articles/712549/fed-circwrestles-with-abstraction-in-network-patent-case</u>> accessed 27 March 2020.

sought to verify the contractual party's ability to pay before the financial transaction would be executed. This computerized process used a trusted third party, an intermediary, who would settle the monetary obligations between the contractual parties of the transactions. The main idea was to prevent the risk of unavailability of the funds to settle the transaction.

To determine whether the patents held by Alice Corp. were patent-eligible, the Court decided to apply the above-mentioned *Mayo* case²⁵ test. There are two steps to be examined: 1) Are the patent claims directed to a patent ineligible concept – laws of nature, natural phenomenon, abstract idea? And if it is so, then the test proceeds to the 2^{nd} step - Does a patent claim involve an inventive concept which makes the abstract idea patent-eligible?²⁶

1.1.3.1 *Alice* Test Step 1. Are the patent claims directed to a patent-ineligible concept – laws of nature, natural phenomenon, abstract idea?

Step one focuses on the judicial exceptions from patentable subject matter. The question is whether Alice Corp.'s patent claims are directed to abstract ideas. "Directed to" does not mean a mere involvement of ineligible concept but, rather, the focus of the claim must be targeted to such concept, as opposed to the improvement of technological process.²⁷ However, the Court has not defined what an abstract idea is. It refused to "delimit the precise contours of the 'abstract idea' category".²⁸ Instead, it provided key steps for identifying what would fall under the category of an abstract idea.

²⁵ Mayo (n 15).

²⁶ Alice (n 7).

²⁷ Hickey (n 16) 16.

²⁸ Alice (n 7) 10.

The Court first noted that the "abstract idea rule" embodies the fact that ideas themselves are not patentable²⁹ and cited *Le Roy v. Tatham* case: "A principle, in the abstract, is a fundamental truth; an original cause; a motive; these cannot be patented, as no one can claim in either of them an exclusive right."³⁰ The Court provided several examples from the previous cases to clarify what would count as an abstract idea:

- 1) *Benson* case³¹: An algorithm itself constitutes to an abstract idea;
- Flook case³²: A mathematical formula which would compute alarm limits and apply it to catalytic conversion processes would also constitute to an abstract idea;
- Bilski case³³: Hedging is a fundamental economic practice which is common in commerce system and thus, it is an abstract idea.

The Supreme Court found that each claim of Alice Corp. was directed to the abstract idea of intermediate settlement.³⁴ Therefore, they were patent-ineligible. More precisely, the claimed patent was a method of hedging against financial risks or price fluctuations and, therefore, constituted to an abstract idea.³⁵ Such method of hedging was "building block of the modern economy" and "method of organizing human activity".³⁶ "The important part of the analysis is to distinguish between the basic building blocks of science and technology and the inventions that integrate them into "something more", such that they satisfy the § 101 eligibility requirement."³⁷ In *Bilski*, The Court stated that abstract ideas are "disembodied concepts" or truths, which are not

²⁹ Ruiz (n 11) 224.

³⁰ Le Roy v. Tatham [1852] 55 U.S. 156, 174–75.

³¹ Gottschalk v. Benson [1972] 409 U.S. 63, 67.

³² Parker v. Flook [1978] 437 U.S. 584, 594-95.

³³ Bilski (n 9).

³⁴ *Alice* (n 7) 15.

³⁵ Ruiz (n 11) 225.

³⁶ *Alice* (n 7) 10.

³⁷ Garza (n 12) 15.

useful from the practical perspective but they could become useful by reducing them to some practical applications.³⁸

Such a vague interpretation of an abstract idea leaves a lot of questions unanswered. If the Court considered the fact of hedging being fundamental economic practice as a decisive clue in this case, then one could easily claim that it adopted the broad definition of abstract idea. As a result, most of the patents can easily be "swallowed" by the abstract idea test because, usually, the inventions are based on fundamental practices - not necessarily economic, but also scientific, biological, financial etc.³⁹ This could hardly have been Court's intention. Thus, some authors have argued that the Court intended to construe this concept of abstract idea in a way to link it to "substantiality" – meaning that, only those claims which are substantially directed to abstract ideas would fall under the first step of the *Alice* test.⁴⁰ This would make more sense as it would prevent most of the patent claims based on fundamental practices from falling under the definition of an "abstract idea". Meanwhile, it would also let the test to proceed to the 2nd prong.

1.1.3.2 *Alice* Test Step 2. Does a patent claim involve an inventive concept which makes the abstract idea patent-eligible?

The 2nd step requires that the claim must involve an inventive concept which makes abstract idea patent-eligible. The Court did not define what exactly constitutes to such concept. The claim has to add "enough" to the abstract idea to transform it into a patent-eligible application. The Court stated that "inventive concept" and "enough" are essentially synonymous as both of them refer to

³⁸ Bilski (n 9).

³⁹ Alice (n 7) 6.

⁴⁰ *Ruiz* (n 11) 225.

elements which ensure that the patent in practice would amount to significantly more than a patent upon ineligible concept – an abstract idea.⁴¹ The purpose of such additional limitations (the Court usually refers to "inventive concept" as "limitations") is to prevent monopolizing the idea itself. Otherwise, the mere abstract idea would become patentable and the conferred monopoly upon it would damage the market. "The mere recitation of a generic computer cannot transform a patentineligible abstract idea into a patent-eligible invention"⁴² – computer must be involved in adding something significant to the value of the claim instead of being a mere medium through which the abstract idea is realized. There should be an element, or combination of elements, which amount to significantly more than the exception itself.⁴³ Thus, "wholly generic" computer involvement is not "enough". If the claimed patent is an abstract idea, the mere fact that it is "applied" by a conventional computer should not be enough to make it patent eligible. In addition to that, conventional, routine or common activities which had already been available and known in the industry would not be enough to transform the claim into patent-eligible claim either.⁴⁴ Merely providing steps which are already known in the art, specified at a high level of generality would not be sufficient enough to confer the "inventive concept" upon the invention.⁴⁵ As an example, the Court indicated that claims in *Alice* did not provide an improvement to the functionality of computers or any other technology or technical field.⁴⁶ So, if software improves the functionality

CEU eTD Collection

 ⁴¹ Ognjen Zivojnovic, 'Patentable Subject Matter After Alice – Distinguishing Narrow Software patents from Overly Broad Business Method Patents' (2015) Berkeley Technology Law Journal Vol. 30:385, 828.
 ⁴² Alice (n 7) 13.

⁴³ United States Patent and Trademark Office, *Revised Patent Subject Matter Eligibility Guidance* (Department of Commerce, Federal Register 50 2019) Vol. 84, No. 4 56 (hereinafter: Revised Patent Subject Matter Eligibility Guidance).

⁴⁴ *Alice* (n 7) 13.

 ⁴⁵ Noam Shemtov, 'Patenting Software, Computer-Implemented Inventions' in *Beyond the Code: Protection of Non-Textual Features of Software* (Oxford University Press 2017) 171.
 ⁴⁶ Alice (n 7) 13.

(computing speed, efficient resource management and etc.), then it may be patentable.⁴⁷ The Court referred to the main idea of the *Benson* case again, which was that the patent claims were ineligible because of the lack of sufficient limitations – they were not limited to any particular art or technology.⁴⁸ If one combines this idea and what the Court stated in *Alice*, the result would be the following: The abstract idea should be limited to a particular art or technology in a way that such a limitation would add "enough" to make the patent-ineligible claim an eligible one. The Court has not determined what should be the language used by the claimant to demonstrate the sufficiency of such limitation either.⁴⁹

Considering all these, the Court stated that the patent in *Alice* did not involve an inventive concept which would transform the claim into patent-eligible one. Therefore, software was not patentable. To shortly summarize the test, John Bednarz provides a comprehensive scheme in his article published on Linkedin.⁵⁰ I will reproduce it on the next page as a conclusion of the *Alice* test:

⁴⁷ Shah IP Law, 'Is Software Patentable in the United States' (2019) <<u>https://www.shahiplaw.com/software-patents/</u>>Accessed 28 March 2020. (hereinafter: Shah IP Law).

⁴⁸ *Benson* (n 31).

⁴⁹ Ruiz (n 11) 227.

⁵⁰ John Bednarz, 'Patentable Subject Matter and Software in 2018' (*Linkedin*, 2018) <<u>https://www.linkedin.com/pulse/patentable-subject-matter-software-2018-john-bednarz</u>> accessed 28 March 2020.



As the analysis has shown, the *Alice* test creates ambiguity beyond the acceptable limit. Patent applicants will find it extremely difficult to understand what they are required to establish in the claims. The further policy implications will be discussed later in the paper. The following section will briefly overview the post-*Alice* attempts of clarifying the law.

1.1.4 After Alice: A Glimmer of Hope for Software Industry

The United States Patent and Trademark Office (USPTO) and several courts have tried to clarify the *Alice* test. This section will briefly show some examples of those attempts.

1.1.4.1 Categories of abstract ideas - USPTO Guidance

Following the Supreme Court's opinion in *Alice*, the USPTO provided guidance regarding the concept of abstract ideas. It examined and combined how the concept has been defined by the different courts and created a sort of categorization:

- "Mathematical concepts mathematical relationships, mathematical formulas or equations, mathematical calculations;"⁵¹
- 2) "Certain methods of organizing human activity fundamental economic principles or practices; commercial or legal interactions (including agreements in the form of contracts; legal obligations; advertising, marketing or sales activities or behaviors;); managing personal behavior or relationships or interactions between people (including social activities, teaching, etc.)";⁵²
- 3) "Mental processes concepts in the human mind (including an observation, evaluation, judgment, opinion)."⁵³ Mental process could be defined as inventions which solve such a problem which can also be solved solely human mind.⁵⁴ Such inventions merely accelerate

⁵¹ Revised Patent Subject Matter Eligibility Guidance (n 43) 52.

⁵² Ibid.

⁵³ Ibid.

⁵⁴ Jay P. Kesan, Carol M. Hayes, 'Patent Eligible Subject Matter After Alice' in John A. Rothschild (eds), *Research Handbook on Electronic Commerce Law* (Edward Elgar Publishing 2016) 244.

the process – what could anyway be solved by human mind without help of computerized means.⁵⁵

The "should" language⁵⁶ used in the text indicates that USPTO attempts to make the list exhaustive, however, considering the rapid development of the patent industry in this field, it allows an exception: In rare cases, when the examiner deems that the claim does not fall within the above-mentioned categories but still considers it to be directed to an abstract idea, the analysis can proceed to the 2nd part of the test.⁵⁷ Therefore, the list is non-exhaustive.

1.1.4.2 When is the claim not directed to an abstract idea? - *Enfish*, *LLC v. Microsoft Corp*. ⁵⁸ example.

Enfish is one of those cases which followed *Alice* and carried the burden of clarifying the 2-step test. The court here struggled with "abstract idea" concept.

The patent in dispute involved a self-referential database. Because of the unique design of the database, it included faster searching of data, more effective storage of data and more flexibility in configuring the database.⁵⁹ Therefore, Court found that the claim here was not directed to abstract idea. Instead, it focused on specific improvement to the way computers operate. It facilitated the computer functionality itself and was not directed to economic or other tasks for

⁵⁵ Garza (n 12) 36.

⁵⁶ "Claims that do not recite matter that falls within these enumerated groupings of abstract ideas should not be treated as reciting abstract ideas". (Revised Patent Subject Matter Eligibility Guidance (n 43) 53).

⁵⁷ Revised Patent Subject Matter Eligibility Guidance (n 43) 53.

⁵⁸ Enfish, LLC v. Microsoft Corp. [2016] 822 F.3d 1327, 118 U.S.P.Q.2d 1684.

⁵⁹ Ladas & Parry LLP, 'Software as Patentable Subject Matter Post Alice' (2016) <<u>https://ladas.com/education-center/5156-2/</u>> Accessed 28 March 2020 (Hereinafter: Ladas & Parry LLP).

which computer could be used generically.⁶⁰ In such cases, claims are considered as patent-eligible at the 1st step of *Alice* test.

Enfish case shows that if patent claim includes a specific improvement to the devices, it satisfies the 1^{st} step of *Alice* test.

1.1.4.3 How do you satisfy step 2 of the Alice test? – DDR Holdings and Bascom examples

The cases below can provide guidance for the 2^{nd} step of *Alice* test. They show examples of how patent-ineligible claim can be transformed into a patent-eligible one.

DDR Holdings v. Hotels.com⁶¹ - An Unconventional Solution

The claim in this case involved "patents directed to systems and methods of generating a composite web page that combines certain visual elements of a "host" website with content of a third-party merchant".⁶² In other words, normally, if the website visitor clicked on the hyperlink, it would take the user to the website of merchant/advertiser. Instead of this, the claimed system would direct the visitor to the composite webpage which retained the host website's "look and feel". The Court stated that while it is difficult to address step 1 of *Alice/Mayo* test, the more important clue is satisfying step 2 by finding an inventive concept.⁶³ It was determined that creation of such "hybrid" website (the idea of which could have been abstract) transformed the claim into patent-eligible one, because such a solution was unconventional.⁶⁴ Taking visitors to a new website would be a conventional, routine solution, while creation of hybrid website was not used in the

⁶⁰ Vic Lin, 'Is software patentable?' (Patent Trademark Blog) <<u>http://www.patenttrademarkblog.com/is-software-patentable/</u> > accessed 28 March 2020.

⁶¹ DDR Holdings v. Hotel.com [2014] (Fed. Cir.) 773 F.3d 1245, 1257.

⁶² Ibid.

⁶³ Mazalin (n 23) 44.

⁶⁴ Shah IP Law (n 47).

prior art. The Court stated that claims did not just recite the performance of practice which was known to the pre-internet world, but the solution here was "necessarily rooted in computer technology" and overcame a problem specific to networks in an unconventional way.⁶⁵

Bascom v. AT&T⁶⁶ – Applying Conventional Components in an Unconventional Way

In this case patent application involved filtering the internet content. It was individually customizable system and targeted at a particular network location. In other words, the filtering tool could be installed remote from users, at a specific location and filtering features could have been chosen by users themselves. The Court found that such solution was unconventional, because Bascom did not claim the filtering system itself but customizable options and remote server for filtering the content. Thus, unique arrangement of otherwise known components in this case amounted to "enough" requirement of the *Alice* step two.⁶⁷ Critical to this determination was focus on the particular arrangement of otherwise generic and conventional components in the claims – because of that, the Court stated that "ordered combination of such limitations" to abstract idea (filtering system) can provide an inventive concept. ⁶⁸

DDR Holdings and *Bascom* cases show that unconventional solution to a network/computer-specific problem can provide an inventive concept and transform patent-ineligible claim into a patent eligible one.

⁶⁵ Garza (n 12) 52.

⁶⁶ Bascom Glob. Internet Servs., Inc. v. AT&T Mobility LLC. [2016] (Fed. Cir.) 827 F.3d 1341.

⁶⁷ Vic Lin (n 60).

⁶⁸ Ladas & Parry LLP (n 59).

1.2 Conclusion

The *Alice* case has drastically affected the number of software patents by significantly reducing them.⁶⁹ Although this case has changed things immensely, software is still patentable in the U.S. However, the law is very unsettled.⁷⁰ Formulation of a test probably, among other things, intended to clarify the law, but *Alice* did the opposite. Lack of any clear guidance on what "abstract idea" stands for, or what amounts to adding "enough" to an ineligible patent-claim, creates a mass ambiguity about software patentability. An ambiguous approach had a very unambiguous result – as mentioned above, significant number of software patents became invalid and new claims are considered patent-ineligible. Since *Alice*, 65% of the patents challenged in district courts have been invalidated, while in the Federal Circuit this number amounts to 91%.⁷¹ Legal uncertainty and lack of predictability may stifle innovation and hinder small businesses to compete in the market.

Even though the balance may have been radically shifted, some still believe that the decision in *Alice* contributed to the fight against the abuse of patent law. Considering the low quality of most software patents before *Alice* (broad and vague claims), patent owners were able to claim that patents covered wide range of technology.⁷² So called "patent trolls" usually bought patents from bankrupt companies and sued their competitors, claiming newly purchased patent's infringement.⁷³ They usually targeted smaller firms and start-ups which do not have enough financial resources to fight back in patent litigation (which are too expensive) and rather choose to settle on a certain amount of money. These actions of patent trolls stifled innovation and software

⁶⁹ Mazalin (n 23) 41.

⁷⁰ Kasdan (n 5) 650.

⁷¹ Gene Quinn, 'What to Do About Alice? Two Years After Ruling, Software patent Eligibility is a Mess' (2016) Eye on Washington 1.

⁷² Electronic Frontier Foundation, 'Deep dive: software patents and the rise of patent trolls' (2013)

<<u>https://www.eff.org/deeplinks/2013/02/deep-dive-software-patents-and-rise-patent-trolls</u>> accessed 19 April 2020. ⁷³ Will Kenton, 'Patent Troll Definition' (Investopedia) < <u>https://www.investopedia.com/terms/p/patent-troll.asp</u>> accessed 30 May 2020.

industry and damaged small businesses which rely on intellectual property assets. Troll's main instruments were exactly those vague and broad patent claims which are being gradually invalidated after *Alice*. Most of the suits were brought by companies which never contributed to the development or innovation and their only intention was to profit from patent infringement suits – threaten the small business which really creates innovative products, sue and demand payment to settle. *Alice* invalidated these low-quality patents and consequently contributed to the fight against patent trolls. Some believe that it has not solved the problem and trolls are going to easily adapt to it.⁷⁴ However, no matter how well *Alice* has struck against such an abuse of patent law, doing it in an under-defined way may instead cause more problems.

Some might even claim that the vagueness of the *Alice* test creates a vast interpretive leeway, which allows courts to decide questions of software patent eligibility in their own discretion.⁷⁵ This claim can hardly have a solid standing in debates, because it cannot compensate the downsides which have terrible impact in terms of legal certainty and predictability. Applicants will face a lot of difficulties in drafting patent applications – how do you prove that your claim is not directed to an abstract idea, when you do not have definition of abstract idea? How do you prove that your claim involves an inventive concept, when you do not know the precise definition of this criteria?

The Supreme Court needs to create a consistent, unambiguous approach to avoid the unreasonable damage to the software patent industry. A patent legislation reform has been planned, targeted at 35 U.S.C. § 101, but it never happened. According to Senator Thom Tillis, any hopes to address the patent eligibility on Congress level have been officially dashed – body does not yet

⁷⁴ James Bessen, 'Why Supreme Court's Alice ruling does not solve patent troll problem' (2014) The Hill.

⁷⁵ Zivojnovic (n 41) 809.

intend to address this problem legislatively.⁷⁶ The only body that tries to clarify the law seems to be the USPTO, guidelines of which are mentioned above. However, they can merely compile what courts provided in different cases and try to assemble the different approaches in a consistent way, which is a difficult task. Therefore, there are two potential solutions – either a Supreme Court clarifying the vague concepts, or legislatively addressing the problem on the Congress level. Otherwise, the harm caused to the software industry may prove to be irreparable.

⁷⁶ Michael Borella, 'The Zombie Apocalypse of Patent Eligibility Reform and a Possible Escape Route' (*Biotech & Pharma Patent Law & News Blog* 2020) <<u>https://www.jdsupra.com/legalnews/the-zombie-apocalypse-of-patent-52573/</u>> accessed 28 March 2020.

Chapter 2 – Patenting Software in Europe

Article 52 of the European Patent Convention (EPC) addresses the definition of patentable inventions in Europe and lists some exclusions. Paragraph one states that: "European patents shall be granted for any inventions, in all fields of technology, provided that they are new, involve an inventive step and are susceptible of industrial application."⁷⁷ This chapter will discuss the issue in the same manner as in the section of the U.S. law. Instead of addressing the original patentability requirements (novelty, inventive step, susceptible of industrial application), it will focus specifically on issues related to software as patentable subject matter. The analysis will rely on the legislation and practice of the European Patent Office (EPO) and its independent judicial body, Boards of Appeal. Boards of Appeal is the second instance which has authority to review the decisions on the grant of patent.

Paragraph 2 of Article 52 EPC lists the exceptions from the patentable subject matter. Subparagraph "c" reads as follows: "schemes, rules and methods for performing mental acts, playing games or doing business, and programs for computers;".⁷⁸ Programs for computers, which basically means software, is excluded patentable subject matter. Does this mean that software can never be patented in European Patent Office? The answer, obviously, is negative. Software can be patented in European Patent Convention even though it is excluded subject matter. Even more, surprisingly, the threshold is not really high as will be discussed below.

⁷⁷ European Patent Convention 1973 Art 52(1) (hereinafter: EPC).

⁷⁸ Ibid. 52(1)(c).

2.1 Computer Programs as Excluded Subject Matter

The European legal tradition has shown that patent protection must be reserved only for those inventions which are technical – so called "technical creations".⁷⁹ This solves the problem caused by the lack of definition of "invention" in EPC. Therefore, the main factor which makes invention patentable and underlies the subject matter is a "technical character". To prove this, it is enough to observe paragraph 2 of Article 52 EPC. If one tried to find similarity between the exclusions listed there, it would be simple to notice that they lack technical character mathematical methods, discoveries, schemes, rules, etc.⁸⁰ This list includes computer programs as well. However, paragraph 3 of article 52 comes into play here and makes these exclusions complicated: "Paragraph 2 shall exclude the patentability of the subject-matter or activities referred to therein only to the extent to which a European patent application or European patent relates to such subject-matter or activities as such."⁸¹ This provision implies that the exclusions must be interpreted narrowly. The wording of the provision ends with "relates to such subjectmatter or activities as such". Therefore, the subject matters listed in paragraph 2 are excluded from patentability if the claims are directed to them as such. However, if invention has a technical character but it is implemented by computer program, it is not excluded subject matter anymore.⁸² Such inventions are referred to as "computer implemented inventions". The requirements for them to be eligible for patent protection remain the same – they must have a technical character. "The expression "computer-implemented inventions" (CII) covers claims which involve computers, computer networks or other programmable apparatus, whereby at least one feature is realized by

⁷⁹ Rainer Osterwalder, 'Patents for software? European law and practice' (European Patent Office 2009) 9.

⁸⁰ EPC (n 77) Art 52(2).

⁸¹ Ibid. 52(3).

⁸² Osterwalder (n 79) 10.

means of a program.⁸³ Thus, the mere fact that an invention is implemented by software does not make it patent ineligible. It is important to note that computer-implemented invention does not involve necessarily only hardware, but software as well. This was established in the *Vicom* case.⁸⁴

2.1.1 Technical Character for Computer-Implemented Inventions

Although there is not a concrete definition of technical character in EPC, Boards of Appeal have provided a lot of clues and tried to clarify its meaning. This section will present the relevant parts of various cases related to the technical character of computer implemented inventions.

Technical character is an implicit requirement set by EPC for the creations to amount to an invention and consequently, qualify for patent protection.⁸⁵ There have been different terms used interchangeably in case law of Boards of Appeal – "technical contribution", "technical effect", "further technical effect", "technical consideration", however, they all relate to the technical character and their threshold has also fluctuated.⁸⁶ Today, the threshold is low. The recent case law has shown that to avoid exclusions under Art. 52(2).c and (3), it is enough to expressly recite the use of a computer, a computer network or computer readable storage medium, because they have technical character.⁸⁷

 ⁸³ European Patent Office, *Examination Guidelines* (2019) <<u>https://www.epo.org/law-practice/legal-texts/html/guidelines/e/f_iv_3_9.htm></u> accessed April 4 2020 (Hereinafter: EPO Guidelines).
 ⁸⁴ Vicom [1986] T 0208/84 (Computer-related invention) EPO Bd. Appeal.

⁸⁵ Pension Benefit Systems Partnership [2000] T 0931/95 EPO Bd. Appeal.

⁸⁶ European Commission, *The trends and current practices in the area of patentability of computer implemented inventions within the EU and the U.S.* (2016) 15 (Hereinafter: European Commission Report).

inventions within the EU and the U.S. (2016) 15 (Hereinatter: European Commission Report).

⁸⁷ Ibid. 16 (citing *Programs for Computers*, G-3/08, [2010] O.J.E.P.O 17, (EPO Enlarged Bd. Appeal).

The European Commission lists several examples of technical character in its report⁸⁸ after reviewing the case law of Boards of Appeal. The examples include:

- Translating natural languages/SYSTRAN (T 1177/97)⁸⁹ The invention was in the field of machine translation of natural languages and concerned so-called "SYSTRAN" translation system. The Board in this case stated that the linguistic information and methods may assume technical character if they are used in a computer system and they solve a technical problem. The use of piece of information in a technical system can confer the technical character to the information itself if it reflects the properties of technical system (This happens when the information is specifically formatted or processed). If technical system uses or processes such information, it may become part of technical solution to a technical problem.
- 2) Data structure product/PHILIPS (T 1194/97)⁹⁰ Invention in this case involved a computer-implemented data structure. Its purpose was to present the visual data in a specific way. A playback device provided fast display of stored pictures using this specific data structure. Board stated that this invention had a technical character, because this specific data structure was implemented only by the playback device and not by human. The interesting issue here is that, originally, "presentation of information" is excluded under EPC Article 52(2), however, as mentioned above, if these exclusions are implemented by technical means (in this case a playback device), then they become patentable.

⁸⁸ European Commission Report (n 86) 15.

⁸⁹ Translating natural languages/SYSTRAN [2002] T 1177/97 EPO Bd. Appeal.

⁹⁰ Data structure product/PHILIPS [2000] T 1194/97 EPO Bd. Appeal.

2.1.2 Computer Programs and "Further Technical Effect"

The threshold for claims which are directed to computer-programs themselves is not as low as it is for computer-implemented inventions. In this case, the computer program must be "capable of bringing about, when running on or loaded on a computer, a further technical effect going beyond the 'normal' physical interactions between the program (software) and the computer (hardware) on which it is run."⁹¹ This is referred to as "further technical effect" approach. It focuses on the functionality of the inventions which go beyond the physical interaction of hardware and software.⁹² The result achieved by the computer program is the important factor here. If the way it affects the computer functionality produces a technical effect directed at solving technical problem, then it is patentable. EPO Guidelines make the definition clear by providing several examples: "a computer program which specifies a method of controlling an anti-lock braking system in a car, determining emissions by an X-ray device, compressing video, restoring a distorted digital image, or encrypting electronic communications"⁹³. "A further technical effect might also be a faster communication between two mobile phones with improved quality of voice transmission".⁹⁴ When these are run on a computer, they provide further technical effect.

2.1.3 Computer-Implemented Inventions versus Computer Programs

As seen in the above sections, EPO considers computer-implemented inventions and computer programs as different things. The relevant decision which addresses their distinction is

⁹¹ European Commission Report (n 86) 16.

⁹² Yannis Skulikaris, 'Patenting Software-related Inventions according to the European Patent Convention A review of past and present law and practice' (2013) European Patent Office.

⁹³ EPO Guidelines (n 83) 3.6.1.

⁹⁴ Osterwalder (n 79) 11.

T 0424/02.95 Paragraph 5.1 of the decision states the following: "Claim 1 relates to a method implemented in a computer system that represents a sequence of steps actually performed and achieving an effect, and not a sequence of computer-executable instructions (i.e. a computer program) which just have the potential of achieving such an effect when loaded into, and run on, a computer. Thus, the Board holds that the claim category of a computer-implemented method is distinguished from that of a computer program. Even though a method, in particular a method of operating a computer, may be put into practice with the help of a computer program, a claim relating to such a method does not claim a computer program in the category of a computer program."⁹⁶ So, computer programs are just a sequence of computer-executable instructions which do not automatically serve purpose of achieving or having any further effects. However, they are still capable of achieving further technical effect as discussed above (Board in this case also stated same: "the computer executable instructions have the potential of achieving the above-mentioned further technical effect of enhancing the internal operation of the computer, which goes beyond the elementary interaction of any hardware and software of data processing").⁹⁷ Meanwhile, computer-implemented inventions are a sequence of steps actually performed and achieving an effect. Although they are also implemented by computer, "a claim relating to such a method does not claim a computer program in the category of a computer program".⁹⁸ This corresponds to the language of EPC article 52.3 – computer programs "as such".

28

CEU eTD Collection

⁹⁵ Clipboard formats I/MICROSOFT [2006] T 0424/03 EPO Bd. Appeal.

⁹⁶ Ibid. 5.1.

⁹⁷ Ibid. 5.3.

⁹⁸ Ibid. 5.1.

In simple words, a computer program is named as such because of the mere fact that it is performed on computer. Meanwhile, computer-implemented invention is an invention which involves a computer and where at least one feature of it is performed by a computer program.

2.1.4 Attempt to Raise the Threshold – Inventive Step and Technical Character

As I've indicated in the introduction, this paper does not focus on the original requirements of patentability but rather, only on software-specific ones. Despite this, the general patentability requirement of involvement of an inventive step cannot be left out from the discussion because it plays a specific role in relation to technical character. Obviously, every invention may have both technical and non-technical features. Any feature of the invention, which is non-technical and therefore excluded from patentability according to Art 52(2) and (3) of EPC, must not be taken into account when existence of inventive step is assessed.⁹⁹ However, this exception does not apply when technical and non-technical features are so connected that they together solve a technical character.¹⁰⁰ An example of *Closing out white space/SAP* case¹⁰¹ will make these clear: Claim involved presentation of visual data on a graphical user interface. The whole inventive step to the invention. The reason was that presentations of information was excluded according to EPC Art 52(2)(d) and therefore, it was a non-technical feature. It did not contribute to the technical character

⁹⁹ European Commission Report (n 86) 17.

¹⁰⁰ Mazalin (n 23) 33.

¹⁰¹ Closing out white space/SAP [2015] T 1562/11 EPO Bd. Appeal.

either. As a non-technical feature, it must not have been taken into account when inventive step was assessed.

This approached was first established in the *Hitachi* case.¹⁰² The claimed invention was a method which carried out automatic auction and this method was computer-implemented. Board stated that as the claim involved technical features, such as server computer and network, it satisfied the subject-matter eligibility. However, Board stated that these claims were not patentable because they lacked inventive step. The problem which was solved by the invention was delays in the propagation of information between the bidders and the server. "The Board found that the closest prior art disclosed an online auction system in which the timing of bids made was included in information sent to the central server handling the auction".¹⁰³ In the present invention, Board found that it solved the problem by adapting the auction method in a way that it would become automatic - both bidders desired price and maximum price were transferred together automatically.¹⁰⁴ Despite this solution, the Boards stated that: "In the Board's view, however, this solution does not contribute to a technical character and cannot therefore be taken into account for assessing inventive step since it concerns the rules of the auction, i.e. it is not a technical solution to the delay problem described (and solved by technical means) [in the cited prior art references], but a solution entirely based on modifications to the auction method. Method steps consisting of modifications to a business scheme and aimed at circumventing a technical problem rather than solving it by technical means cannot contribute to the technical character of the subject matter claimed".¹⁰⁵ So, the introduced technical features of the invention were already known in the prior

¹⁰² Auction method/HITACHI [2004] T 0258/03 EPO Bd. Appeal.

¹⁰³ European Commission Report (n 86) 19.

¹⁰⁴ Ibid.

¹⁰⁵ Ibid. (citing *Hitachi* case (n 102)).

art and they did not have inventive step. What was inventive here, was rules of auction, which was non-technical. Therefore, they must not have been considered.

2.2 Conclusion

The discussion above shows that the language used in EPC Article 52(2)(c) is sort of formality and can be easily overcome. Software in Europe is indeed patentable, however, the claimants should be cautious during drafting the patent claims and they should take into account the relevant legislation and practice. The higher the formal requirements are, the greater care is required. The difference between computer programs and computer-implemented inventions, technical character and unique approach towards the assessment of inventive step are those crucial features which have to be considered.

Chapter 3 Comparison and Implications

Software is patentable in both U.S. and European jurisdictions. However, there are limitations regarding the subject matter eligibility. Thresholds and key definitions are different, however, several elements of the Court's reasoning in *Alice* resemble the EPO approach. One can even claim that the idea, the most important rationale underlying the limitations is the same. Lack of technical character is something both jurisdictions unanimously consider as decisive. U.S. legislation/case law does not explicitly mention it, however, the three judicial exceptions – laws of nature, natural phenomenon – all lack the technical character and that could have been idea the judges had in mind when they excluded them from patentability. The Supreme Court in Alice stressed the fact that the claims did not provide an improvement to the functionality of computers or any other technical field. This is similar to the "further technical effect" standard of EPO. Both approaches focus on the improvement of functionality as one of the important factors of patentability. "Computer programs as such" and "substantially directed to abstract idea", "technical character", "further technical effect" and "enough" of Alice/Mayo are also similar standards. The assessment manner may not be completely same but the rationale behind them is similar. While the U.S. tries to exclude overly broad patents by introducing a rule which refuses claims substantially directed to abstract ideas, EPO achieves the same by introducing "as such" wording. A patent application claiming computer program as such in EPO would be similar to a patent application claiming a computer process substantially directed to the abstract idea in the U.S. A feature which adds "enough" to the abstract idea to make it patent eligible in the U.S., would most probably provide "further technical effect" for a conventional computer program in EPO patent claim.

Similarity of the standards can be explained by the tricky nature of software. In this industry innovators are particularly dependent on their competitors' inventions. Software is a field where several components can be used together to achieve a certain result. New entrants to the market often rely on the methods developed by others to build up their businesses. Overly broad patents can endanger innovation. This becomes particularly sensitive in this field where hundreds of software are being developed every day and often their functional aim can be similar. Claiming this functionality in patent applications will create overly broad patents, resulting in detriment of innovation. The U.S. Supreme Court addressed this issue to the probably highest extent possible, by narrowing down the subject matter in relation to software. EPO also addresses the issue in the same manner, however, its approach is less strict.

The paragraph above can further lead to a delicate topic of whether less stringent patent protection contributes to innovation and competition. Some argue that free access to the methods developed by competitors enhances the innovation. Others retort that stripping small businesses of software patent protection means weakening their starting assets and making them less competitive. This controversy is out of the scope of this paper. Notwithstanding its importance, discussing it cannot be the very first issue in the list of priorities needed to be addressed by competent authorities. The reason is simple – one cannot argue about IP policy implications when the IP policy itself does not provide certainty. Whether or not high patentability thresholds benefit competition, creating such limitation in an underdefined way is definitely not a feasible solution. Unless the protection offered by legal framework becomes clear, its further consequences remain obscure.

Due to the more formal approach, the EPO standards are clearer and more precise than the U.S. standards. The latter does not have a consistent approach and has undergone a lot of

significant changes which have resulted in a mess and confusion. Legal certainty is of an utter importance and the U.S. lacks it beyond acceptable limit. Such obscurity is a disaster for inventors and those enterprises which rely on their intellectual property assets. As mentioned above, the number of software patents is unreasonably declining in recent years. This could have been possible to justify if it happened due to the consistent, clear approach of USPTO. However, the only reason behind this is ambiguity. Again, this paper does not intend to criticize less stringent approach or vice versa. If clearer standards will be introduced eventually, if might turn out that the enormous number of patent claims were improperly denied. Instead, if the new approach will be even stricter then, maybe, it will turn out that the claims were indeed properly denied. The point is that the ambiguity at present endangers the industry as a whole. The appropriateness of the denials is less critical as far as they exist because of obscurity.

The paragraph above criticizes the U.S. system because of ambiguity but this should not lead the reader to consider EU approach as consistent. From the comparative perspective, EU seems clearer but considering it from an objective point of view, it indeed lacks the certainty as well. First of all, the exclusion in EPC Art 52(2)(c) itself causes confusion in the first place. It is an unnecessary complication of the language of the provision. The first impression after reading the provision is that computer program is an excluded subject matter. It could lead to a wrong understanding that software cannot be patented in Europe. The wording of Art 52(3) is not really clear – meaning of "*as such*" could be subject to wrong interpretations. Realizing the importance of this paragraph, a clearer approach would have been more than welcome. This leads to the unnecessary differentiation between computer-implemented inventions and computer programs which is a bit ambiguous without the further research of literature and Boards of Appeal cases.

One could notice that ambiguity in Europe exists in formal part – wording of provisions and terms. On opposite, in the U.S. the ambiguity exists in the substantive part of law. In other words, EPO knows what is patentable and what is not. If there is any difficulty, that is because of the lack of consistency in the organization and systemization of the law. However, in the U.S., one can claim that it is still ambiguous what kind of software is patentable. In addition to the vagueness, the frequent change of approach makes this problem even more critical.

Different approaches of such major software-innovator jurisdictions can lead to the problems on international level. Software industry is a field without territorial limitations. The larger this industry becomes, the less blurry is the boundaries of its use in different countries. Innovative software product of a certain start-up somewhere in Eastern Europe may become target of large tech firm from the U.S. and vice versa. Tech giants from the U.S. may want to apply for patent protection far beyond the borders of the U.S. and again, vice versa. International harmonization regarding the issue of software patentability is crucial. The Patent Cooperation Treaty (PCT) allows international patent applications but in the field of software it may cause specific problems. Software product may qualify for patent protection in EPO, but USPTO may find it difficult to recognize the patent granted by its European "colleague". Considering the potential this industry provides for the surge in economic growth worldwide, it would be a grave mistake to undermine this potential because of the lack of harmonization between the jurisdictions. Such jurisdictions where large part of world's software innovators is located need to resemble their standards. Similar approaches would contribute to the legal certainty on the global level.

Even though the future changes in the software industry may not be foreseeable yet, one thing is clear – software start-ups will find European legal environment more attractive. Could this lead to the migration of software patent claims from the U.S. to Europe? Answer to this question

does not depend merely on legal framework considerations. Tech/software start-ups will take into account the business environment, potential of success, implications of the economic systems of the countries where they plan to establish. The following years will make it clear whether the *Alice* standard will remain as a fuzzy judicial test. Due to the effort of the courts in the subsequent decisions, the contours of the framework might become clearer. However, judging from today's perspective, even though there is a glimpse of hope, it is not yet clearly foreseeable or promising. If a certain and predictable legal environment is all that matters for an average software start-up which is about to enter the market, its choice should be Europe.

An ever-expanding software industry needs to find the most appropriate regulatory environment to flourish. The U.S. and Europe need to start re-considering their policies. What would amount to a perfect regulatory framework? When it comes to the field of various innovations, the question of regulation always leads to controversial topics. Controversies become particularly sensitive in the software industry due its unique features, as discussed above. However, there are two main aims any authority should consider when regulating this field: 1) ensuring certainty and predictability; 2) ensuring incentives to research and innovation while not deterring dissemination and access. The question may come up whether there is a priority in addressing these two issues. The answer could be affirmative – the former might be more important because of two reasons. First, because addressing the balance between promoting incentives to innovation and ensuring dissemination needs a well-defined legal framework. Second, because without well-defined coordinates, any discussion of the viability of a regulatory framework is rendered moot. However, these two issues still are not clearly separable and they might intertwine. Both must be taken into account to the highest extent possible. Instead of separation, they could indeed amount to one large process – ensuring certainty *while* providing both incentives and access.

The U.S. may want to consider the reasons behind EPO's more consistent approach. The cooperation between EPO and Boards of Appeal (which together make one large authority) greatly contributes to creating a consistent legal environment. Meanwhile, in the U.S., the Congress, USPTO and judicial system being separate bodies may result in lack of necessary cooperation in creating a comprehensive framework. Expert testimonies cannot always fill the gap created by the lack of specific technical knowledge required for the courts to decide software-related matters. This may lead to problems. Denmark, for example, addressed such problem by creating a centralized patent court - Copenhagen Maritime and Commercial Court. Being a specialized court in IP matters and using combination of legal and "technical" judges (experienced specifically in patent law) significantly contributed to the creation of a certain legal environment.¹⁰⁶ This could be one of the examples to follow for the U.S. It will be more complicated to accomplish in a much larger system, but it could prove to be a reasonable solution. Software patents is a field which requires vast amount of technical knowledge and this issue should be addressed properly. Decisions in this field impact not only IP framework, but also business environment, competition etc. Thus, it needs proper attention by those authorities which are well-aware of what is at stake. Courts specialized specifically in patent law may solve the existing issues.

In conclusion, factors that various legal systems need to consider when addressing this field at the minimum are the following: certainty, predictability, balance between ensuring incentives to research/innovation and enabling access/dissemination, international harmonization. Taking

¹⁰⁶ Susanne Høiberg, Louise Aagaard, 'Improving legal certainty in patent cases' (2013) Building and enforcing intellectual property value 83.

into account the first two factors, European approach is superior. It is not perfect. Even more, it is far from the modern standards of certainty required in different systems of law. However, compared to the U.S. standards, it is more consistent.

Conclusion

The rapid expansion of the software industry leaves us with no choice but to alter the perception of what human mind alone can achieve. Software has become part of almost every single aspect of our lives. One can hardly remember if world's legislators have ever faced a tougher challenge than regulating phenomena introduced by the era of technologies. They have to either create new laws or try to adapt the existing ones to the changes. It seems that in the field of IP, the U.S. and European IP offices have chosen the latter – adjustment of existing laws. The following years will show whether director general of WIPO is right in claiming that existing IP laws are not reliable for software industry anymore. However, patents and copyrights remain as the most common choices of developers to protect their software inventions so far.

Software is patentable in both Europe and the U.S. The Supreme Court in the U.S. established test-based approach in *Alice* case and it checks whether the patent claim is directed to an abstract idea and if it involves an inventive concept which makes the abstract idea patent-eligible. Lack of definitions of the relevant concepts of the test creates ambiguity which harms the software industry. Applicants cannot predict the outcome of their patent applications. Legal uncertainty undermines the potential this industry offers. The U.S. needs to consider the necessity of introducing proper legislation by the Congress and the potential benefits of creating a centralized patent court.

EPO addresses software patent issues on legislative level. However, there as well is a slight vagueness regarding the several relevant concepts in determining patentability of the claims. Boards of Appeal has tried to make things clear in several cases and they have obviously achieved more success than the U.S.

This paper concludes that EPO has established a superior approach compared to the U.S. in terms of consistency. It offers more certain and predictable environment. If a comprehensive legal framework is more important than other considerations for the entrants in the new market, then they will find Europe as a more attractive place. Test-based approach of the U.S. undoubtedly has some advantages, including the flexibility and contribution to the fight against patent trolls. However, adoption of an under-defined standard in the software industry is a grave mistake which undermines its prosperous potential.

Changes in the future are inevitable. At some point, both systems will indeed realize the necessity and importance of much larger attention this field requires. Considering the speed this industry evolves at, we may soon reach the point when existing IP rights simply will not be capable of being applied to the software innovations. However, at present, the priority is quite clear – creation of the most facilitating environment for the software industry to flourish. First steps that need to be taken must be towards the improvement of certainty and predictability. Legal systems need to decide what exactly they offer. While deciding this, they must ensure to provide incentives to innovation while not deterring the access and dissemination. They need to consider the importance of international harmonization to contribute to the certainty on the global level. The opportunities offered by the software industry are limitless. States must realize what is at stake. Software is present. Software is future. It needs proper legal attention.

Bibliography

Primary Sources

Legislation

35 U.S.C. § 101 European Patent Convention 1973

U.S. Cases

Alice Corp. v. CLS Bank International [2014] 573 U.S. 208
Bilski v. Kappos [2010] 561 U.S. 593
In re Bilski [2008] (Fed. Cir.) 545 F.3d 943
Bascom Glob. Internet Servs., Inc. v. AT&T Mobility LLC. [2016] (Fed. Cir.) 827 F.3d 1341
DDR Holdings v. Hotel.com [2014] (Fed. Cir.) 773 F.3d 1245
Enfish, LLC v. Microsoft Corp. [2016] (Fed. Cir.) 822 F.3d 1327, 118 U.S.P.Q.2d 1684
Gottschalk v. Benson [1972] 409 U.S. 63
Le Roy v. Tatham [1852] 55 U.S. 156, 174
Mayo Collaborative Servs. v. Prometheus Labs. Inc. [2012] 132 S. Ct. 1289
Parker v. Flook [1978] 437 U.S. 584, 594

European Cases

Auction method/HITACHI T 0258/03 [2004] EPO Bd. Appeal Clipboard formats I/MICROSOFT, [2006] T 0424/03 EPO Bd. Appeal Closing out white space/SAP [2015] T 1562/11 EPO Bd. Appeal Data structure product/PHILIPS [2000] T 1194/97 EPO Bd. Appeal Pension Benefit Systems Partnership [2000] T 0931/95 EPO Bd. Appeal Translating natural languages/SYSTRAN [2002] T 1177/97 EPO Bd. Appeal Vicom [1986] T 0208/84 EPO Bd. Appeal

Secondary Sources

- American Bar Association, 'A Virtual Analog Rule for Software Patent Eligibility' (*Landslide Magazine*, 2015) <<u>https://www.bradley.com/insights/publications/2015/11/a-virtual-analog-rule-for-software-patent-eligib</u> > accessed 28 March 2020
- Bednarz J, 'Patentable Subject Matter and Software in 2018' (*Linkedin*, 2018) <<u>https://www.linkedin.com/pulse/patentable-subject-matter-software-2018-john-bednarz</u>> accessed 28 March 2020
- Bessen J, 'Why Supreme Court's Alice ruling does not solve patent troll problem' (2014) The Hill
- Borella M, 'The Zombie Apocalypse of Patent Eligibility Reform and a Possible Escape Route' (*Biotech & Pharma Patent Law & News Blog* 2020) <<u>https://www.jdsupra.com/legalnews/the-zombie-apocalypse-of-patent-52573/</u>> accessed 28 March 2020
- BSA Foundation, 'Software: Growing US Jobs and the GDP' (2018) <<u>https://www.software.org/reports/software-growing-us-jobs-and-the-gdp/</u>> accessed 8 May 2020
- Chao B, 'Finding the point of Novelty in Software patents' (2013) Berkeley Technology Law Journal, Vol. 28:1217
- Clyde C, 'Current Issues in Technology and Law' (2015) <<u>https://www.mckercher.ca/blog/current-issues-in-technology-and-the-law</u>> accessed 15 April 2020
- Electronic Frontier Foundation, 'Deep dive: software patents and the rise of patent trolls' (2013) < <u>https://www.eff.org/deeplinks/2013/02/deep-dive-software-patents-and-rise-patent-trolls</u>> accessed 19 April 2020
- European Patent Office, Examination Guidelines (2019)
- European Commission, *The trends and current practices in the area of patentability of computer implemented inventions within the EU and the U.S.* (2016) 15 (Hereinafter: European Commission Report)
- Garza R, 'Software Patents and Pretrial Dismissal Based on Ineligibility' (2018) 24 Rich JL & Tech 1
- Hickey K, 'Patent-Eligible Subject Matter Reform in the 116th Congress' (2019) Congressional Research Service, R4591

- Høiberg S, Aagaard L, 'Improving legal certainty in patent cases' (2013) Building and enforcing intellectual property value
- Kasdan A, 'Can You Patent Software and Business Methods in the U.S. How Did We Get Here and Where Do We Now Stand' (2014) 24 Fed Cir BJ 64

Kenton W., 'Patent Troll Definition' (Investopedia) <<u>https://www.investopedia.com/terms/p/patent-troll.asp</u>> accessed 30 May 2020

- Kesan J, Hayes C, 'Patent Eligible Subject Matter After Alice' in John A. Rothschild (eds), Research Handbook on Electronic Commerce Law (Edward Elgar Publishing 2016)
- Ladas & Parry LLP, 'Software as Patentable Subject Matter Post Alice' (2016) <<u>https://ladas.com/education-center/5156-2/</u>> Accessed 28 March 2020
- Lin V, 'Is software patentable?' (Patent Trademark Blog) <<u>http://www.patenttrademarkblog.com/is-software-patentable/</u> > accessed 28 March 2020
- Mazalin T, 'A comparative analysis of the patentability of software in the U.S. and Europe' (Master thesis, Leibniz Universität Hannover 2017)
- Macagnone M, 'Fed. Cir. Wrestles with Abstraction in Network Patent Case' (*LAW360*, 2015) <<u>http://www.law360.com/articles/712549/fed-circwrestles-with-abstraction-in-network-patent-case></u> accessed 27 March 2020
- Osterwalder R, 'Patents for software? European law and practice' (European Patent Office 2009)
- Quinn G, 'What to Do About Alice? Two Years After Ruling, Software patent Eligibility is a Mess' (2016) Eye on Washington
- Ruiz C, 'Patentable Software: Analyzing Alice Under a law-and-Economics Framework and Proposing a new Approach to Software Claims', (2017) B.U. J. SCI. & TECH. L, Vol. 23:208,
- Shah IP Law, 'Is Software Patentable in the United States' (2019) <<u>https://www.shahiplaw.com/software-patents/</u>> Accessed 28 March 2020
- Skulikaris Y, 'Patenting Software-related Inventions according to the European Patent Convention A review of past and present law and practice' (2013) European Patent Office
- Shemtov N, 'Patenting Software, Computer-Implemented Inventions' in *Beyond the Code: Protection of Non-Textual Features of Software* (Oxford University Press 2017) 171
- United States Patent and Trademark Office, 'Revised Patent Subject Matter Eligibility Guidance' (Department of Commerce, Federal Register 50 2019) Vol. 84, No. 4

- Westermann H, 'How to treat software in the intellectual property framework' (Bachelor thesis, Lund university 2016)
- WIPO Magazine, 'Francis Gurry on the future of intellectual property: opportunities and challenges', 2017
- Zivojnovic O, 'Patentable Subject Matter After Alice Distinguishing Narrow Software patents from Overly Broad Business Method Patents' (2015) Berkeley Technology Law Journal Vol. 30:385