The Current Dilemma and Future of Software Patenting

Yahong Li*

Introduction

On 19 June 2014, the US Supreme Court decided in *Alice v. CLS Bank* ('Alice') that computer software consisting of an abstract idea is not eligible for patent protection.[1] The impact of this decision is still profound and present after five years of the ruling. For example, by mid-2018, over 400 court decisions have found the software patents invalid in the United States.[2] Assessments of the impact have been mixed. Some have celebrated *Alice* because under its ruling, 'bad patents went down, and software innovation went up',[3] patent trolls have been curbed or reduced, and software related R&D and employment have witnessed a huge growth.[4] Others have lamented that the decision 'has perpetuated a sense of uncertainty [and]… prompted many to cast a grim prospect for the software patent industry'.[5] Intellectual Property Owners Association and American Bar Association have lobbied to amend US Patent Act section 101 in an effect to overturn *Alice* decision.[6] The divisions over *Alice's* impact have stemmed from the ambiguities contained in the Court's 'abstract idea' and 'significantly more' construct, based on the so-called two-step test invented in the *Mayo v. Prometheus* case.[7] The two-step test in *Mayo* requires the determination of patent eligibility to undergo the following two-step analysis: (1) whether a patent claim recites a law of nature or an abstract idea. If so, then (2) whether the claim has added something

* JSD and JSM (Stanford), Associate Professor and Director of LLM in Intellectual Property and Information Technology Law, Faculty of Law, the University of Hong Kong. The early draft of this article was presented at the PKU-Stanford-Oxford Internet Law and Public Policy Conference held on November 23, 2014, when *Alice* case was just decided. The author is grateful for Professor Paul Goldstein and the other conference participants' comments on my presentation. The article has been updated with the research assistance of Mr. Wang Chuandao. The author thanks the anonymous reviewers for their critical and insightful comments and suggestions which helped strengthen this article in many aspects. All the errors, however, are author's own.

[1] *Alice Corporation PTY. Ltd. v CLS Bank International et al*. 573 U.S. 208, 134 S. Ct. 2347 (2014).
[2] These decisions include both US federal courts' decisions and the decisions by the Patent Trial and Appeal Board. See Daniel Nazeer (2018), 'Happy Birthday Alice: Four Years Busting Software Patents', Electronic Frontier Foundation, 22 June 2018, https://www.eff.org/deeplinks/2018/06/happy-birthday-alice-four-years-busting-software-patents
[3] *Ibid*.
[4] *Ibid*., also see, Electronic Frontier Foundation, 'Saved by Alice', https://www.eff.org/alice; Martin Goetz, 'Why Alice v. CLS Bank is a Victory for Software Patents', IPWatchDog, http://www.ipwatchdog.com/2015/02/06/alice-v-cls-bank-is-a-victory-for-software-patents/id=54489/
[5] Babak Nouri (2018), 'A Realistic Perspective on Post-Alice Software Patent Eligibility', IPWatchDog, 14 October 2018, https://www.ipwatchdog.com/2018/10/14/realistic-perspective-post-alice-software-patent-eligibility/id
[6] Lee Rawles, 'Erase Ambiguity in Patent Eligibility Law, ABA IP Section Urges USPTO', 7 April 2017, ABA Journal, http://www.abajournal.com/news/article/erase_ambiguity_in_patent_eligibilty_law_to_encourage_tech_growth_aba_ip_se
[7] *Mayo Collaborative Services v Prometheus Laboratories, Inc*., 132 S. Ct. 1289 (2012).

substantially more to transform the "law of nature" or "abstract idea" into an eligible patent subjective matter.[8] Although *Mayo* concerns a method of administering drug to patients, not a computer program, it was heavily relied upon by the US Supreme Court in reaching the decision for *Alice*,[9] and has been widely cited by many other post-*Mayo* patent cases, whether or not about computer programs.[10] The ambiguity about "abstract idea" or "law of nature" has been joined by the argument that software (or a computer program[11]) should not be the subject matter of a patent because it is a 'statement', 'program language' or 'mathematic algorithm' that is under copyright protection.[12] A similar ambiguity and debate have existed elsewhere in the world, e.g., the unclear and much debated concepts of "computer program *as such*", technical "effect", "contribution" and "character" in European patent regime; and the concepts of "computer program *per se*" and "technical solution" in Chinese patent regime. These concepts are in turn muddled with patentability criteria such as novelty and inventiveness, making the situation worse. The ambiguities and arguments concerning software patents have caused much confusion and uncertainty in patent examinations and court decisions, and undermined patent protection for software innovation. They also show a dilemma that the software patent regime is facing: whether the regime should be strengthened with more relaxed eligibility and patentability criteria, or weakened by the more stringent concept of "abstract idea"? Either way could have strong implications to software innovation.

Recently, patent offices in the US, Europe and China have amended their patent examination guidelines in an attempt to clarify the ambiguities concerning software patent eligibility.[13] In particular, the US Patent and Trademark Office ('USPTO') has amended the Patent Subject Matter Eligibility Guidance to provide a grouping of "abstract ideas" and the steps to decide whether some of the "abstract ideas" could be transformed into patentable software.[14] This article will conduct a historical and comparative overview of software patenting in the US, Europe and China, and discuss the recent amendments to the patent examination guidelines as to software patent eligibility in these three jurisdictions. Based on these overview and discussions, this article argues that, in addition to finding technical solutions by national patent offices, a more holistic solution should be sought, ideally, at the international level as to the problems of software patenting because they are the ones faced by many countries. Such solution should address not only the issue of software patent eligibility, but also the issue of designing a patent regime in which valuable software inventions can be granted patents fairly easily but the scope and term of these patents are limited so that the true innovations can be incentivized and patent trolls can be prevented.

---

[8] *Ibid*.
[9] See Alice case, note 1 above.
[10] For example, *Association for Molecular Pathology v. Myriad Genetics, Inc.*, 569 U.S. 576 (2013); *Ariosa Diagnostics, Inc. v. Sequenom, Inc.*, 788 F.3d 1371 (Fed. Cir. 2015); Genetic Technologies Ltd. v. Merial L.L.C. (Fed. Cir. 2016); Cleveland Clinic Foundation v. True Health Diagnostics LLC (Fed. Cir. 2017).
[11] In this paper, the terms software and computer program are interchangeable.
[12] Some have argued that software should not even be protected by copyright because 'copyright is not supposed to be about functionality'. Paul Goldstein's statement in an interview by John Roemer, 'Copyright's Highway Take 2', *Stanford Lawyer*, Issue 98/Vol. 52/No. 2, at 34.
[13] See the discussion in Part II, section 2.
[14] The revised guidance was released on 4 January 2019, see USPTO, 2019 Revised Patent Subject Matter Eligibility Guidance, https://s3.amazonaws.com/public-inspection.federalregister.gov/2018-28282.pdf, see the discussion in Part II, section 2.1

This article is divided into two parts. Part I takes a historical approach to overview software patenting in three major jurisdictions: the US, Europe and China, to present the dilemma of software patenting faced by national patent authorities, courts, legal practitioners, and software industries in these three jurisdictions. The focus of this part is on the pre and post Alice development in the US, the conflicting rulings in Europe, and the inconsistency between law and practices in China. In part II, the article comparatively analyses the dilemma presented in part I, and takes a holistic view on software patenting by discussing four issues that are beyond *Alice* or the issue of 'abstract idea': (1) Whether patent is necessary for software innovation? (2) How have the concepts of 'abstract idea', 'computer program as such', or 'computer program per se' been defined in the US, Europe and China and how should they be better defined? (3) What is a proper length for software protection? (4) What are the proposed solutions and what should be a better solution for software patenting. Based on the foregoing, the article concludes that patent is still needed for software innovation, but the scope and length of protection should be limited to promote more innovation. The courts and patent offices around the world should not be constrained by the approach adopted in *Alice* decision, but should rather strive harder to provide clearer criteria of eligibility for software patents, which go beyond 'abstract idea', or 'computer program as such or per se' mindset, and finally consider adopting a more holistic solution at international level.

Part I. Software Patenting Dilemma in the US, Europe and China from a Historical Perspective

Software patenting has been plagued with uncertainty since its birth. To assess what is the effect of the uncertainty of the judicial rulings on the future of software patenting, it is useful to revisit the history of software patenting in the US, Europe and China, which represent three major patent regimes having a large volume of patents.

1. Software patenting in the United States

1.1 The Pre-Alice Era

Software patenting before *Alice* has been through several notable stages in the United States, from software being rejected as an unpatentable abstract idea in the 1970s to gradually being accepted as a patentable computerised invention in the 1980s under the 'machine-or-transformation' test, to its peak in the 1990s under the 'useful, concrete and tangible result' test, and then its gradual decline in the 2000s when the courts returned to the 'machine-or-transformation' test, and finally in the 2010s, to a totally uncertain and dilemmatic situation in which the courts delivered inconsistent and contradictory decisions under the 'abstract idea' plus 'significantly more' constructs under *Alice*.

The first software patent was granted to Martin Goetz in 1965.[15] By 1972, the US Supreme Court had invalidated a software patent in *Gottschalk v. Bensen.*[16] It held

[15] Marty Goetz, 'Pioneer Goetz rebuts software patent critics', ZDNet, http://www.zdnet.com/article/pioneer-goetz-rebuts-software-patent-critics/, accessed 14 Dec. 2017.
[16] *Gottschalk v Bensen* 409 U.S. 63 (1972).

that a new information processing method used in a computer system to translate the representation of numerical data from one recording format to another was not patentable. In its view, the software involved a 'mathematical algorithm', and patenting it would pre-empt others from using that algorithm.[17] Similarly, in the 1978 decision in *Parker v. Flook*, the Supreme Court found that a computer program for adjusting an alarm setting was a mathematical algorithm and not a process within the meaning of patent law.[18]

This trend took a drastic turn in the 1980s, starting with *Diamond v Diehr.*[19] In that case, the Supreme Court adopted the 'machine-or-transformation' test articulated in *Gottschalk v Bensen.* In so doing, it held that a computer program performing a curing process for rubber qualified as a process for the purpose of securing patent protection so long as it performed more than mathematical calculations. In other words, the software 'as such' was not patentable, but the software was patentable if it involved a physical machine or was a process making use of a mathematical algorithm and 'transforming or reducing an article to a different state or thing'.[20] After *Diamond v Diehr*, more software patents were granted by the US Patent and Trademark Office ('USPTO') and upheld by the US courts.[21]

Software patenting reached its peak in the 1990s, fuelled by the CAFC's very generous test of 'useful, concrete and tangible'. Under this test, a computer program was patentable if it produced a 'useful, concrete and tangible result' even if it did not involve a machine or any transformation. Thus, in *In re Alappat*, the CAFC held that a program controlling the waveform on a screen was patentable because it evaluated electronic signals and determined how to best display the signals on a cathode-ray tube screen.[22] This ruling was immediately followed by *In re Lowry,* in which a data structure representing information on a computer's hard drive was deemed a patentable device.[23] Following *In re Alappat*, the USPTO issued roughly13,500 software patents in 1997 alone.[24] Many business methods-related software patents were also granted. For example, a 'data processing system for hub and spoke financial services configuration' (e.g. mutual fund pooling) was held patentable in the famous case of *State Street Bank v Signature.*[25] A program allowing a '1-click' virtual check out for Amazon's online payment system was also found to be patentable in *Amazon v Barnesandnoble*.[26]

---

[17] *Gottschalk v Bensen*, *ibid*.
[18] *Parker v Flook* 437 U.S. 584 (1978).
[19] *Diamond v Diehr*, 450 U.S. 175 (1981).
[20] *Ibid*.
[21] For example, in *Paine et al v Merrill Lynch et a*l., 564 F Supp. 1358 (1983) a patent was awarded for a computer program allowing data processing of the financial management system.
[22] *In re Alappat* 33 F. 3d 1526 (1994).
[23] *In re Lowry*, 32 F. 3d 1579, 1582 (Fed. Cir. 1994)
[24] Anthony Williams (2005), The Patent Explosion, http://anthonydwilliams.com/wp-content/uploads/2006/08/The_Patent_Explosion.pdf
[25] *State Street Bank & Trust Co. v Signature Financial Group, Inc*., 149 F. 3d 1368 (1998). The CAFC held as follows: 'Today, we hold that the transformation of data, representing discrete dollar amounts, by a machine through a series of mathematical calculations into a final share price, constitutes a practical application of a mathematical algorithm, formula, or calculation, because it produces "a useful, concrete and tangible result"—a final share price momentarily fixed for recording and reporting purposes and even accepted and relied upon by regulatory authorities and in subsequent trades'.
[26] *Amazon.com v Barnesandnoble.com*, 239 F. 3d 1243 (2001). The District Court held that the 1-click program was patentable, but this decision was reversed by the CAFC because prior art for the program had been found. After CAFC's decision, the parties reached a settlement in early 2002.

However, the trend for software patenting took another twist in the 2000s. The US courts seemed to have been awakened by the flood of low quality software patents and began to pull back. In *In re Bilski*,[27] the CAFC abolished the 'useful, concrete and tangible result' test and returned to the 'machine-or-transformation' test. In so doing, it invalidated the patent for a method of hedging risks associated with a commodity sold at a fixed price on the ground that the method claimed an abstract idea. Although the Supreme Court disagreed that the 'machine-or-transformation' test should be the sole test,[28] it failed to provide an alternative. As a result, after *Bilski* the courts' decisions on software patenting went hither and yonder. For example, a method and system for detecting credit card fraud in Internet transactions was held unpatentable,[29] whereas a method for monetising and distributing copyrighted products over the Internet was patentable.[30] Thereafter, the unprecedented split among the CAFC judges in the *Alice* case occurred, as discussed earlier.[31] The post-Alice trend will be addressed after we have surveyed the situations in Europe and China.

1.2 A Recap of Alice

Alice v. CLS bank involved patents for a computer implemented scheme to mitigate 'settlement risk' by using a third-party intermediary. The patents were assigned to Alice and their validity and enforceability were challenged by CLS. In return, Alice filed a lawsuit alleging infringement. The patents had several claims include: (1) a method for exchanging financial obligations; (2) a system to carry out the method; (3) a computer-readable medium containing program code to perform the method. The central issue was whether these claims were patent eligible.[32]

The US District Court for the District of Columbia held that all of the claims were patent ineligible because they were directed to the abstract idea of 'employing a neutral intermediary to facilitate simultaneous exchange of obligations in order to minimise risk'.[33] The US Court of Appeals for the Federal Circuit ('CAFC') reversed the District Court's decision, holding that it was not 'manifestly evident' that the claims were directed to an abstract idea.[34] However, this panel opinion was vacated in a rehearing *en banc*. Writing for the five-member (out of 10) plurality, Judge Lourie concluded that all of the claims were patent ineligible, confirming the District Court's decision that they 'draw on the abstract idea of reducing settlement risk by effecting trades through a third-party intermediary', and that 'the use of a computer to maintain, adjust, and reconcile shadow accounts added nothing of substance to that abstract idea'.[35] On 19 June 2014, the US Supreme Court affirmed the CAFC decision, holding that 'the claims at issue are drawn to the abstract idea of intermediated settlement, and that merely requiring generic computer implementation fails to transform that abstract idea into a patent-eligible invention'.[36]

---

[27] *In re Bilski* 545 F. 3d 943 (2008).
[28] *Bilski v Kappos*, 561 U.S. 593 (2010). The Supreme Court said that the 'machine-or-transformation' test shall not be the sole test of patentability although it is 'a useful and important clue … for determining whether some claimed inventions are processes under s. 101'.
[29] *CyberSource Corp v Retail Decisions, Inc.* (Fed. Cir. 2011).
[30] *Ultramercial, LLC v Hulu, LLC*, 722 F. 3d 1335 (Fed. Cir., 2013).
[31] See part 1 above.
[32] *Alice Corp. v CLS Bank International*, *ibid.*, at 1.
[33] *Alice Corp. v CLS Bank International*, 768 F. Supp. 2d 221, 252 (DC 2011).
[34] *Alice Corp. v CLS Bank International*, 685 F. 3d 1341, 1352, 1356 (2012)
[35] *Alice Corp. v CLS Bank International*, 573 U.S. 208, 134 S. Ct. 2347 (2014), at 4.
[36] *Ibid.*, at 1.

The case attracted wide public attention not only because its outcome had a profound impact on software patenting, but because the CAFC was unprecedentedly divided in delivering its decision. At first, the panel divided by 2-1 reversed the District Court decision. Then it granted rehearing *en banc* and ordered the parties and any *amici curiae* to address several issues concerning the patentability of the claims.[37] A 10 member panel eventually vacated the earlier panel's opinion and affirmed the District Court's judgment in a one-paragraph *per curiam* opinion. Among the 10 CAFC judges on the panel, however, the views were seriously fractured. Seven judges found the method and media claims to be patent ineligible. The court *en banc*, affirming the District Court's ruling, found by an equally divided vote that the system claim was not patent eligible. As a result, no single opinion was supported by a majority on all points, and no single test for the patentability of the computer-implemented invention was adopted by the court. The court had to issue a plurality opinion.[38]

The five-member plurality concluded that all of the claims were patent ineligible. However, this view was opposed by the other judges in their separate opinions. The then Chief Judge Rader wrote a separate opinion (joined by Moore) in part agreeing with the plurality that the method and media claims were patent ineligible; and in part (joined by Linn, Moore and O'Malley) that the system claims were patent eligible. Judge Moore wrote a separate opinion dissenting in part, arguing that the system claims were patent eligible. Judge Newman wrote a separate opinion concurring in part and dissenting in part, arguing that all of the claims were patent eligible. Judges Linn and O'Malley wrote another separate dissenting opinion arguing that all of the claims were patent eligible.

It is even more interesting to note that, in contrast to the CAFC and unlike its decisions in other controversial patent cases, the US Supreme Court unanimously invalidated the *Alice* patent without any reservations. This sent a clear signal that the it was not wavering in its recent trend to halt software patenting, and particularly computer-implemented business method patenting. The Supreme Court adopted the two-step test in *Mayo v. Prometheus*[39] to determine whether the *Alice* claim was directed to a patent-eligible subject matter under §101. The first step was to decide whether the claim at issue was directed to an unpatentable law of nature, natural phenomenon or an abstract idea. If so, the second step would be to decide whether any element or combination of elements in the claim amounted to significantly more than a patent on the law of nature itself, that is, an inventive concept.[40] The court held that 'the claims at issue are drawn to the abstract idea of intermediated settlement, and that merely requiring generic computer implementation fails to transform that abstract idea into a patent-eligible invention'.[41]

The battle in the *Alice* case, both substantively and procedurally, has demonstrated that the debate on what software is and whether it can be patented has reached its climax. Software patenting has come to a crossroads that leads to all directions and uncertainties. A clearer and more holistic solution is urgently needed.

---

[37] Specifically, there are three issues to be addressed: (1) what test should the court use to decide whether a computer-implemented invention is a patent ineligible abstract idea; (2) whether the presence of a computer in a claim could ever make patent-ineligible subject matter patentable; and (3) whether method, system, and media claims should be considered equivalent under § 101.

[38] A plurality opinion does not mean a majority opinion, but the opinion that receives more support than any other opinion.

[39] *Mayo Collaborative Services v Prometheus Laboratories, Inc*., 132 S. Ct. 1289 (2012).

[40] *Mayo v Prometheus*, *ibid.,* p 4.

[41] *Alice Corp. v CLS Bank International et al*. 573 U.S. 208, 134 S. Ct. 2347 (2014), p 1.

1.3 Post-Alice Trend

1.3.1 Conflicting rulings after Alice

After *Alice*, hundreds of patents were invalidated on the ground that the claims were based on unpatentable abstract ideas. For example, *Alice* was followed by the CAFC rendering yet another contradictory opinion in *Ultramercial v Hulu.* In this case, the CAFC panel, led by the then Chief Judge Rader, twice held Ultramercial's software patent to be patentable. However, the panel, led by Judge Mayer (after Judge Rader retired) reversed the earlier rulings,[42] and held otherwise in its third ruling after being instructed by the Supreme Court to follow *Alice*.[43] In *Two-Way Media v Comcast Cable,* both the District Court and the CAFC applied *Alice's* two-step test, holding that the claims were directed at the abstract ideas of sending, directing, monitoring and accumulating information, which 'merely invokes generic processes and machinery' without further inventive concepts. Therefore, the claims were patent ineligible.[44] *Smart Systems v Chicago Transit Authority* was another case in which the Federal Circuit affirmed the District Court's decision, holding that the asserted claims required no more than generic computer implementation, and therefore they had not ventured into the territory of §101 eligibility.[45]

However, there has been a growing trend for the lower courts to resist *Alice* and find ways to hold software inventions patentable. For example, in *Enfish LLC v Microsoft*, the District Court invalidated the patent, holding 'the patents' claim a 'logical table' demonstrates abstractness'.[46] The CAFC reversed, holding that the self-referential table 'is a specific type of data structure designed to improve the way a computer stores and retrieves memory'. Thus, it deserved a patent.[47] In the 2012 decision of *Amdocs v Openet Telecom,* Israel-based Amdocs sued an Irish company in the US courts, alleging infringement of its patents for online accounting and billing methods. The District Court held that the plaintiff's claim was not much more than an abstract idea correlating two networks. The CAFC reversed, holding that '[T]his claim entails an unconventional technological solution (enhancing data in a distributed fashion) to a technological problem (massive record flows that previously required massive databases)'. Similarly, in *McRo v Bandai Namco Games America,* the District Court held the patent invalid because it claimed an abstract idea and would have pre-empted any lip synchronisation that used a 'rules-based morph target approach'. The CAFC reversed, holding that 'the claimed improvement here is allowing computers to produce "accurate and realistic lip synchronisation and facial expressions in animated characters", which could previously only have been produced by human animators'. BSA (Business Software Alliance) president Victoria Espinel, commented on this case:

> The Federal Circuit's opinion reaffirms that software is worthy of patent protection just as any other field of technology. Software is a major component of today's greatest innovations, and it is imperative that our patent system continues to encourage innovators in all fields of technology. Today's Federal Circuit's decision is a step in the right direction.

---

[42] *Ultramercial, LLC v. Hulu, LLC*, 722 F. 3d 1335 (Fed. Cir, 2013).
[43] *Ultramercial, LLC v Hulu, LLC,* No. 10-1544 (Fed. Cir. 14 Nov. 2014).
[44] *Two-Way Media Ltd. v Comcast Cable Communications, LLC* (Fed. Cir. 2017).
[45] *Smart Systems Innovations, LLC v Chicago Transit Authority*, 873 F. 3d 1364 (Fed. Cir. 2017), at 19.
[46] *Enfish, LLC v. Microsoft Corp.*, 56 F. Supp. 3d 1167, 1176 (C.D. Cal. 2014).
[47] *Enfish LLC v Microsoft Corp.,* (2016) U.S. App. LEXIS 8699, 2016 WL 2756255 (Fed. Cir. 12 May 2016).

1.3.2 The difficulties in following Alice

Undoubtedly, *Alice* has been a watershed for software patenting in US patent history. During the post-*Alice* era, the courts in most cases have attempted to follow the *Alice* framework in terms of subject matter eligibility, but they have struggled to apply this framework to specific cases. This has been demonstrated in several ways.

First, absent a statutory definition of 'abstract idea', the courts have struggled under step one to determine whether a claimed software invention is directed to an abstract idea. In *Enfish*,[48] the Federal Circuit admitted the absence of a definitive rule to 'determine what constitutes an "abstract idea" sufficient to satisfy the first step of the *Mayo/Alice* inquiry' and observed that the existing practice was to compare 'claims at issue to those claims already found to be directly to an abstract idea in previous cases'.[49] The court attempted to provide guidance in this regard. For example, it contended that for a 'directed to' inquiry, the claim must be assessed as a whole in the context of its specifications.[50] In *McRO*,[51] the court stressed that the claim must be 'considered in its entirety',[52] but it did not stop there. It further stated that claim construction was helpful for assessing patent eligibility under § 101. Moreover, in *Enfish*,[53] the court created what was later called a specific improvement test by asking 'whether the focus of the claims is on the specific asserted improvement in computer capabilities'.[54] The specific improvement test was adopted recently in the step one inquiry in both *Finjan*[55] and *Core Wireless Licensing.*[56]

Second, the step two inquiry under *Alice* has also presented a great challenge to the courts. It has required the presence of something significantly more, that could transform the abstract idea into a patent eligible subject matter. The term 'significantly more', however, is general and vague. The court must decide on a case by case basis whether the claimed invention has something significantly more than an abstract idea. In *DDR Holdings*, the court confirmed the presence of something significantly more because the claimed solution was 'necessarily rooted in computer technology in order to overcome a problem specifically arising in the realm of computer networks'.[57] In *Content Extraction*, the court held that the claimed invention must involve performance more than 'well-understood, routine, [and] conventional activities previously known'.[58] In *BASCOM*, the court held that a claimed invention could pass a step two inquiry if it improved an existing technological process.[59] In *Electric Power Group*, the court objected to patent eligibility based on merely limiting the claims to a particular technological environment.[60] In *Secured Mail Solutions*, the court held that merely applying an abstract idea to a general computer failed to pass the step two test.[61] Similar rulings were reached in *Inventor Holdings*.[62]

---

[48] *Enfish, LLC v Microsoft Corp.,* 822 F.3d 1327 (Fed. Cir. 2016).
[49] *Ibid.*, at 9.
[50] *Ibid.*, at 10.
[51] *McRO Inc. v Bandai Namco Games America Inc.* Case 15-1080 (Fed. Cir. 2016).
[52] *Ibid.,* at 19.
[53] *Enfish, LLC v Microsoft Corp.,* 822 F.3d 1327 (Fed. Cir. 2016).
[54] *Ibid.,* at 11.
[55] *Finjan Inc v Blue Coat Systems Inc.,* 879 F. 3d 1299 (Fed. Cir. 2018).
[56] *Core Wireless Licensing S.A.R.L., v LG Electronics. Inc.,* 880 F. 3d 1356 (Fed. Cir. 2018).
[57] *DDR Holdings, LLC v Hotels.com, L.P.,* 773 F.3d 1245 (Fed. Cir. 2014), at 20.
[58] *Content Extraction v Wells Fargo Bank,* 776 F.3d 1343 (Fed. Cir. 2014), at 8.
[59] *Bascom Global Internet Services, Inc. v AT&T Mobility, LLC*, 827 F.3d 1343 (Fed. Cir. 2016), at 17.
[60] *Electric Power Group LLC v Alstom S.A.* 830 F.3d 1350 (Fed. Cir. 2016), at 8-9.
[61] *Secured Mail Solutions LLC v Universal Wilde, Inc.* 873 F. 3d 905 (Fed. Cir. 2017), at 10.
[62] *Inventor Holdings, LLC v Bed Bath & Beyond, Inc,* 876 F.3d 1372 (Fed. Cir. 2017), at 11.

The case law gives the impression that clear rules are present in the step two test. This is untrue. For example, it was established as early as *Mayo* that claim limitations involving merely 'well-understood, routine, conventional activity previously engaged in by scientists in the field' were not patent eligible.[63] However, the meaning of 'well-understood, routine, conventional activity' has been unclear. In *Berkheimer v. HP Inc.,* the CAFC held that '[t]he question of whether a claim element or combination of elements is well-understood, routine and conventional to a skilled artisan in the relevant field is a question of fact', and such fact 'must be proven by clear and convincing evidence', and not appropriate for a summary judgment.[64] The court added that the disclosure of prior art was not necessarily well-understood, routine or conventional.[65] This ruling was petitioned by HP Inc. to Supreme Court on 28 September 2018 and is now pending for Supreme Court to decide 'whether patent eligibility is a question of law for the court based on the scope of the claims or a questions of fact for the jury based on the state of the art at the time of the patent'.[66]

1.3.3 The confused §101 inquiry and its overlapping with other sections

Admittedly, the post-*Alice* era has provided rich case law guidance regarding eligibility for software patenting. However, the guidance has been insufficient, inconsistent, and oftentimes confusing. The most problematic issue has been the confusing signals sent by the courts when they have referred to sections other than §101 to assess patent eligibility for software claims. In *Diehr*, the Supreme Court found that '[t]he 'novelty' of any element or steps in a process, or even of the process itself, is of no relevance in determining whether the subject matter of a claim falls within the §101 categories of possibly patentable subject matter'.[67] That is, a §101 inquiry is independent of and separate from §102 and §103 inquiries. However, this rule has not been satisfactorily followed. For example, step two in *Alice* has been described as a search for an 'inventive concept'.[68] Yet the words, 'inventive concept' as used therein are misleading. Thus, step two resembles the obviousness analysis under §103, especially considering that the courts have provided no clear definition of what should constitute an inventive concept that imparts something significantly more to a software claim. Indeed, even the Supreme Court has struggled with this matter. In *Mayo,* on one hand, the Court found that the overlap between §101 and §102 inquires was sometimes necessary 'in evaluating the significance of additional steps'. On the other hand, it objected to shifting the §101 inquiry entirely to §102 and §103 inquires because this would create significant legal uncertainty.[69] Worse yet, the Court did not elaborate on under what circumstances and to what extent such an overlap was justifiable.

The overlap between §101 and §102 or §103 has been observed in case law. In *BASCOM*, the court reasoned that the claimed way of filtering content represented a technical improvement over prior art and could pass the step two test.[70] In *Electric Power Group*, the court held that the claim at issue contained no 'inventive set of components or methods' and it affirmed the District Court's decision that the claimed

---

[63] *Mayo v Prometheus, 566 U.S. 66* (2012), at 2-3.
[64] *Berkheimer v HP Inc.* 881. F.3d 1360 (Fed. Cir. 2018), at 12.
[65] *Ibid.,* at 14.
[66] HP Inc. v. Berkheimer, Petition for a Writ of Certiorari, at i, https://www.supremecourt.gov/DocketPDF/18/18-415/65216/20180928162630738_36823%20pdf%20Hong%20I%20br.pdf
[67] *Diamond v Diehr* 45 U.S. 175 (1981), at 7.
[68] *Alice Corp. v CLS Bank International,* 573 U.S. 208, 134 S. Ct. 2347 (2014), at 7.
[69] *Mayo v Prometheus,* 566 U.S. 66 (2012), at 21.
[70] *Bascom Global Internet Services, Inc. v AT&T Mobility, LLC* 827 F.3d 1343 (Fed. Cir. 2016), at 16.

inventions were patent ineligible.[71] In *Secured Mail Solutions*, the technologies embodied in the claims were treated as well-known and involving no inventive concept.[72] Thus, wording such as 'technical improvement', 'inventive' and 'well-known' have presented the impression that some claims should fall within the territory of §102 and §103.

The courts seem to have noticed this issue and have attempted to address the uncertainty. However, their efforts have sometimes exacerbated rather than alleviating the uncertainty. For example, in *BASCOM*, the court contended that it did not matter whether each element in a claim had been known in the art. The inventive concept was present if the elements had been arranged in a non-conventional and non-generic way.[73] The court's reasoning suggests that when evaluating patent eligibility under §101, prior art must be searched. The claim elements can then be compared with the prior art to evaluate whether each element is known and the combination is non-conventional and non-generic, which, in substance, resembles the novelty inquiry under §102. However, there has been no clear definition of what is regarded as 'non-conventional and non-generic.' It seems that a combination, even known in the art, may still be considered non-conventional and non-generic under certain circumstances. As a result, in the courts' view, searching for an 'inventive concept' under *Alice s*tep two goes beyond a §102 inquiry. Alternatively, such searching does not extend beyond a §103 inquiry. An *Alice* step two inquiry stops to the extent that it does not consider whether the 'inventive concept' is obvious to a person skilled in the art.[74] As a result, the courts seem to have created a new statutory bar to the patentability test, lying somewhere between §102 and §103.

Another issue has resulted from the distinction between the abstract and non-abstract parts of a claim. *Alice* step two concentrates on whether the non-abstract part represents an inventive concept.[75] This is not as easy as it appears because the two parts are not always clearly distinguishable.[76] This also contravenes the holding of the Supreme Court that a claim must be constructed as a whole.[77]

The third issue relates to the specific improvement test. In *Enfish,* the court found that the claimed self-referential table embodied a specific improvement to the operation of computers, and therefore was not directed to an abstract idea under *Alice* step one.[78] In *Finjan*, the court held that the claimed behaviour-based virus scanning method constituted a specific improvement to computer functionality, and therefore could pass step one.[79] In *Core Wireless Licensing*, the court held that the claims at issue were directed to the improved user interface of electronic devices, such as computers, and involved a specific improvement over prior art systems, thereby surviving step one.[80] In these cases, the court applied the specific improvement test under step one. However, the *BASCOM* court treated the test differently, holding that the specific

---

71 *Electric Power Group LLC v Alstom S.A.* 830 F.3d 1350 (Fed. Cir. 2016), at 9.
72 Sec*ured Mail Solutions LLC v Universal Wilde, Inc.* 873 F. 3d 905 (Fed. Cir. 2017), at 12.
73 *Bascom Global Internet Services, Inc. v AT&T Mobility, LLC* 827 F.3d 1343 (Fed. Cir. 2016), at 15.
74 Ted G. Dane, 'Part I: Are the Federal Circuit's Recent Section 101 Decisions a "Specific Improvement" in Patent Eligibility Law'? (2018) 35(3) Computer and Internet Lawyer 1-16, at 8.
75 Ted G. Dane, 'Part II: Are the Federal Circuit's Recent Section 101 Decisions a "Specific Improvement" in Patent Eligibility Law'? (2018) 35(4) Computer and Internet Lawyer 5-15, at 7.
76 Ted G. Dane, 'Part I: Are the Federal Circuit's Recent Section 101 Decisions a "Specific Improvement" in Patent Eligibility Law'? (2018) 35(3) Computer and Internet Lawyer 1-16, at 10.
77 *Diamond v Diehr* 45 U.S. 175 (1981), at 6.
78 *Enfish, LLC v Microsoft Corp.,* 822 F.3d 1327 (Fed. Cir. 2016), at 12.
79 *Finjan Inc v Blue Coat Systems Inc.*879 F. 3d 1299 (Fed. Cir. 2018), at 6.
80 *Core Wireless Licensing S.A.R.L., v LG Electronics. Inc.* 880 F. 3d 1356 (Fed. Cir. 2018), at 9.

improvement test was also applicable under step two.[81] This was confusing and self-contradictory: If a specific improvement is unable to prevent the claimed invention from being directed to an abstract idea under step one, how is it capable of transforming the abstract idea into patent eligible matter under step two?

2. Software Patenting in Europe

Software patenting in Europe has a history of uncertainty as well. There has been no uniformed legal definition for 'programs for computers', except for an explanation under the Guidelines for Examination.[82] Uncertainty over how to assess the patent eligibility of software inventions has also been reflected in the following three approaches: the technical contribution approach, technical effect approach, and any hardware approach.[83] During the evolution of these approaches, the boards and courts have struggled to determine the proper way to address software eligibility. In principle, Article 52 of the European Patent Convention ('EPC') regards 'computer programs as such' as unpatentable subject matter. However, what 'as such' means has been a wild guessing game for the courts. Before 2006, a program was not 'as such' and could be patentable if it had made a 'contribution' or had a 'technical effect' in the field.[84] After 2006, this test has shifted and a computer program was patentable if it simply had a 'technical character'.[85] Nonetheless, in practice what constitutes 'technical character' has had different interpretations. Sometimes the court has required a 'further technical effect'[86] to be demonstrated, and in others the mere claim of a technical means has been enough.[87]

2.1 'Technical contribution' approach

The technical contribution approach evolved from the landmark decision in the 1986 *Vicom*[88] case. It was later replaced by the technical effect approach in the 1990s.[89]

Vicom involved claims directed to a method of digitally processing images and related apparatuses. The European Patent Office ('EPO') Board of Appeal ('board') held that '[a] claim directed to a technical process, which process is carried out under the control of a program (whether by means of hardware or software), cannot be regarded as relating to a computer program as such' and '[a] claim which can be considered as being directed to a computer set up to operate in accordance with a specified program (whether by means of hardware or software) for controlling or carrying out a technical process cannot be regarded as relating to a computer program

---

[81] Ted G. Dane, 'Part I: Are the Federal Circuit's Recent Section 101 Decisions a "Specific Improvement" in Patent Eligibility Law'? (2018) 35(3) Computer and Internet Lawyer 1-16, at 9.
[82] EPO 2018 Guidelines for Examination, 'Index for Computer-Implemented Inventions (CII)' defines a CII as 'one which involves the use of a computer, computer network or other programmable apparatus, where one or more features are realised wholly or partly by means of a computer program.' (G-II, 3.6) See https://www.epo.org/law-practice/legal-texts/html/guidelines/e/j.htm.
[83] Rosa Maria Ballardini,' Software Patents in Europe: The Technical Requirement Dilemma' (2008) 3(9) Journal of Intellectual Property Law & Practice, 563-575, at 565-566.
[84] T 52/85.
[85] T 258/03 and T 154/04. Also see European Patent Office, Guidelines for Examination, Article 3.6, paragraph 4.
[86] See T1173/97.
[87] See T 424/03, and EPO Guidelines for Examination, Article 3.6, paragraphs 3 and 7.
[88] T 0208/84
[89] Rosa Maria Ballardini,' Software Patents in Europe: The Technical Requirement Dilemma' (2008) 3(9) Journal of Intellectual Property Law & Practice, 563-575, at 565.

as such'.[90] The board emphasised that '[t]he appellants have thus made a new and valuable contribution to the stock of human knowledge and patent protection for this contribution cannot be denied merely on the basis that the manner in which the invention is defined would appear to bring it within the exclusions of Articles 52(2) EPC'.[91] When it comes to patentability, in particular, the board stated '[d]evisive is what technical contribution the invention as defined in the claim when considered as a whole makes to the known art'.[92] This was the first time 'technical contribution' had been expressly considered, and the holding also suggested that known art must be consulted when determining the technical contribution.

The *Vicom* decision was later applied and developed in *Koch and Sterzel*[93] and *Text processing/IBM*.[94] In *Koch and Sterzel,*[95] the holding supported the patentability of the claimed invention because it contributed to a field that was not excluded from patentability. In *Text processing/IBM*,[96] the findings rejected the patentability of the claimed method at issue because it 'merely makes use of a computer program, running on conventional hardware' and nothing 'could be regarded as making a contribution to the art'. The board clearly stressed that 'the intention of the EPC to permit patenting in those cases in which the invention involves some contribution to the art in a field not excluded from patentability'.[97]

Ballardini pointed out that the technical contribution approach requires examiners to consult prior art to determine whether an inventive step can be justified for the claimed inventions.[98] However, this lacks legitimacy and creates ambiguity because an inventive step is a separate and independent threshold under the EPC.[99] Logically, patent eligibility should be tested before looking at the inventive step, and patent eligibility should be evaluated independently without the need to refer to other tests. It is further submitted that the technical contribution approach has probably resulted in subjective conclusions because whether there is a technical contribution depends largely on the prior art searched. This varies significantly depending on the search skills and strategies of the searchers.

2.2 'Technical effect' approach

The technical effect approach prevailed from the 1990s to around 2000,[100] but has ceased to be effective at the EPO, although it is still applied in the UK.

In T0935/97, the board stated '[d]etermining the technical contribution an invention achieves with respect to the prior art is therefore more appropriate for the purpose of examining novelty and inventive step than for deciding on possible exclusion under Article 52(2) and (3)'.[101] However, the board further ruled that 'a

---

[90] T 0208/84
[91] *Ibid.*
[92] *Ibid.*
[93] T0026/86, Koch and Sterzel, 1987
[94] T0038/86, Text processing/IBM, 1989
[95] T0026/86, Koch and Sterzel, 1987
[96] T0038/86, Text processing/IBM, 1989
[97] *Ibid.*
[98] Rosa Maria Ballardini,' Software Patents in Europe: The Technical Requirement Dilemma' (2008) 3(9) Journal of Intellectual Property Law & Practice, 563-575, at 565.
[99] Article 56 EPC.
[100] Lionel Bently and Brad Sherman, Intellectual Property Law, 4th Ed (Oxford: Oxford University Press, 2014), at 455.
[101] T0935/97 – 3.5.1

computer program claimed by itself is not excluded from patentability if the program, when running on a computer or loaded into a computer, brings about, or is capable of bringing about, a technical effect which goes beyond the 'normal' physical interactions between the program (software) and the computer (hardware) on which it is run'. This was cited in T1173/97.[102] Under these rulings, it is allowable to claim a computer program product, such as a computer-readable medium that carries computer programs. It does not matter that the basic idea underlying the software resides in the computer program itself rather than in the computer program product.

2.3 'Any hardware' approach

The present practice of evaluating patent eligibility at EPO has shifted to the 'any hardware' approach, which evolved from the decision in *Pension Benefit Systems Partnership*,[103] and later was largely expanded by *Hitachi*.[104] Under this approach, an invention does not fall within the list of exclusions in Article 52(2) EPC so long as it 'embodies or is implemented by some technical means (such as a computer)',[105] regardless of how well-known and mundane the technical means is.[106]

Pension Benefit Systems Partnership [107] concerned a European patent application claiming a method and an apparatus for controlling a pension benefits program. By administering at least one subscriber employer account on behalf of each subscriber employer's enrolled employees, each was to receive periodic benefits payments. The board treated the method claims and apparatus claims differently. More specifically, for method claims, it concluded that '[m]ethods only involving economic concepts and practices of doing business are not inventions within the meaning of Article 52(1) EPC'. However, for apparatus claims, the board applied the 'any hardware' approach and ruled that '[a]n apparatus constituting a physical entity or concrete product suitable for performing or supporting an economic activity, is an invention within the meaning of Article 52(1) EPC'.

In addition, the board emphasised what was stated in the guidelines, that 'the basic test of whether there is an invention within the meaning of Article 52(1), is separate and distinct from the questions whether the subject-matter is susceptible of industrial application, is new and involves an inventive step'.[108] Accordingly, the board refused to apply the 'technical contribution' test, stating that it 'confuses the requirement of "invention" with the requirements of "novelty" and "inventive step"'. The board also considered the 'technical contribution' approach to be unjustifiable, stating that '[t]here is no basis in the EPC for distinguishing between "new features" of an invention and features of that invention which are known from the prior art when examining whether the invention concerned may be considered to be an invention within the meaning of Article 52(1) EPC. Thus, there is no basis in the EPC for applying this so-called contribution approach for this purpose'. Further, according to the board, to evaluate whether a computer program had technical character, the correct question

---

[102] T1173/97 -3.5.1
[103] T0931/195 Controlling Pension Benefit System/PBS Partnership, 2001.
[104] T0258/03, Hitachi/Auction method, 2004.
[105] Lionel Bently and Brad Sherman, Intellectual Property Law, 4th Ed (Oxford: Oxford University Press, 2014), at 465.
[106] Ewan Nettleton, 'Software Patentability Ruling from the European Patent Office's Enlarged Board' (2010) 17(3-4) Journal of Database Marketing & Customer Strategy Management, 267-270, at 268.
[107] T0931/195 Controlling Pension Benefit System/PBS Partnership, 2001.
[108] EPO Guidelines for Examination 2018, Part G –Cap II-2, para. 2.

to ask was 'whether the claim involves the use of or is to a piece of physical hardware, however mundane. If it is, Art. 52(2) does not apply'.[109]

Hitachi[110] involved a European patent application claiming an automatic auction method executed in a server computer, a computerised auction apparatus for performing an automatic auction via a network and a computer program, which ran on a computer network comprising client computers and a server. Unlike *Pension Benefit Systems Partnership*, the board in *Hitachi* applied the any hardware approach to both the method claims and the apparatus claims. Notably, it ruled that 'in general, a method involving technical means is an invention within the meaning of Article 52(1) EPC'. This was in stark contrast to *Pension Benefit Systems Partnership,* which supported '[a] feature of a method which concerns the use of technical means for a purely non-technical purpose and/or for processing purely non-technical information does not necessarily confer a technical character to such a method'. *Hitachi* therefore suggested that a claimed invention, so long as it contained technical means, such as a computer, would not fall within the meaning of Article 52(2) EPC regardless of whether the claimed invention was for a purely non-technical purpose.

The any hardware approach lowered the threshold for patent eligibility for software inventions. As Kitchin J said in *Astron Clinica*, 'the board appears to have found that any program on a carrier has a technical character and so escapes the prohibition in Article 52 following Hitachi'.[111] However, this did not necessarily mean that more software inventions would be patented because the claimed invention still needs to pass other tests, such as novelty and inventive step test.

2.4 The current EPO patent framework

Under the current EPO patent framework, to become an eligible subject matter, a claim for a computer programs must be 'technical' or have a 'technical character'.[112] This is implied in the EPC Implementing Regulations, according to which the claimed invention must be related to a technical field (Rule 42(1)(a)), be concerned with a technical problem (Rule 42(1)(c)), and possess technical features (Rule 43(1)). The board in *Pension Benefit Systems Partnership*[113] held that 'technical character is an implicit requirement of the EPC'; and that any invention, not merely a software invention, must have a 'technical character' to be patentable.[114] However, as stated above, a statutory definition is still lacking as to what constitutes a 'technical character' that can prevent a claimed software invention from being treated as a program for computers *as such*.

This situation has led to inconsistent interpretations and legal uncertainty, resulting in conflicting rulings. For example, in *T1173/97 (Microsoft)*, the Technical Board of Appeal (TBA) focused on the function of the program rather than the manner in which it was claimed. It found that a claim to a computer program was patentable if it had 'technical character', without distinguishing whether the technical effect was 'direct' or 'indirect' and whether the claim was for the computer program itself or for

---

[109] Rosa Maria Ballardini,' Software Patents in Europe: The Technical Requirement Dilemma' (2008) 3(9) Journal of Intellectual Property Law & Practice 563-575, at 566.
[110] T0258/03, Hitachi/Auction method, 2004.
[111] *Astron Clinica v Comptroller General of Patents*, [2008] R.P.C. 14 339-355, at 351.
[112] G- II, 3.6, Guidelines for Examination in the European Patent Office (November 2017).
[113] T0931/195 Controlling Pension Benefit System/PBS Partnership, 2001.
[114] Sigrid Sterckx and Julian Cockbain, 'The Patentability of Computer Programs in Europe: An Improved Interpretation of Articles 52(2) and (3) of the European Patent Convention' (2010) 13(3) Journal of World Intellectual Property 366-402, at 368.

a carrier.[115] However, in *T424/03 (IBM)*, the TBA distinguished between claims pertaining to the methods implemented in computer systems and claims to the 'computer program' itself, focusing on the manner in which the program was claimed. It determined that claims for the 'computer program' were excluded, but computer implemented methods and computer products were not.[116] To solve the problem of the conflicting rulings, in October 2008, the EPO President referred the question of computer program patentability to the EPO Enlarged Board of Appeal (EBA). The EBA found that the referral was 'inadmissible' and refused to decide it, saying that '*when judiciary-driven legal development meets its limits, it is time for the legislator to take over*' [*emphasis added*].[117] In the meanwhile, it has attempted to provide clear guidance through the following cases.

In *T0424/03*, the EBA held that '[a] computer system including a memory (clipboard) is a technical means, and consequently the claimed method has technical character in accordance with established case law'; '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'; and '[t]he claimed method modifies the internal operation of a computer system and is therefore technical'. This decision suggests that 'enhancing or modifying the internal operation of the computer' goes beyond the normal effect and can thus be treated as a 'further technical effect', making it patent eligible.

In *T0121/06*, claim 9 pertained to a program for trash pin. According to decision *T1173/97*, point 13 (OJ EPO 1999,609), a computer program is not a program *as such* (cf. Article 52(2), (3) EPC) if it is capable of achieving a 'further' technical effect. The EBA held that the program for trash pin achieved a 'further' technical effect because it served to free up memory space that would otherwise be unnecessarily blocked and hence it modified the internal functioning of the computer itself. This went beyond the 'normal' technical effects, i.e., the physical interactions between the computer program and computer memory, which any program involves. Further, it was consistent with the decision in *T0424/03*, finding that modifying the internal functioning of the computer imparted the software invention's further technical effect.

In *T0318/10*, the EBA held: 'A basic difference between a mathematical method and a technical process can be seen in the fact that a mathematical method is carried out on numbers and provides a result also in numerical form, the mathematical method being only an abstract concept prescribing how to operate on the numbers and producing no direct technical result as such. In contrast thereto, if a mathematical method (in the present case the optimization function) is used in a technical process (in the present case assigning a server from among a plurality of servers to process a client request), that process is carried out on a physical entity by some technical means implementing the method and provides as its result a certain change in that entity. In the present case, the optimisation function as defined by claim 1 results in a change in the resource consumption of the server system which is a technical effect.' This suggests that a computer program, by embodying a mathematical method, can become

---

[115] In this decision, no 'direct' or 'indirect' technical effect was distinguished, nor was the computer program claimed by itself or as a record on a carrier.

[116] EPO Enlarged Board Referral on Software Patents, 24 October 2008, http://ipkitten.blogspot.hk/2008/10/epo-enlarged-board-referral-on-software.html.

[117] Opinion of 12 May 2010, EPO Enlarged Board of Appeal, Case No. G 0003/08, http://documents.epo.org/projects/babylon/eponet.nsf/0/DC6171F182D8B65AC125772100426656/$File/G3_08_Opinion_12_05_2010_en.pdf.

patent eligible when executing the computer program results in a certain change to a physical entity.

In *T1568/05*, it was stated that, '[t]he Board acknowledges the technical character of the method according to claim 1 because the method uses *technical means* (a computing device) for a *technical* purpose in a distributed file system'; and '[t]he computer-executable instructions according to claim 38 have the potential for achieving the aforementioned technical effect which goes beyond the elementary general interaction between software and hardware'. This suggests that using a technical means for a technical purpose could put the claimed software invention in a better position for patent eligibility. The bottom line is that the interaction between the software and hardware must go beyond an 'elementary general interaction'. Compared with manual calculations (such as calculating the circular constant, pi) using a pen and paper, using a computer is generally more efficient and accurate. To be patent eligible, the claimed software invention must go beyond such an elementary general interaction.

In *T1351/04*, the following explanation was provided: 'Functional data, intended for controlling a technical device, are normally regarded as having technical character. One example of this is decision T 110/90 - Editable document form/IBM (OJ EPO 1994,557), in which 'control items' (e.g., 'carriage return') were regarded as having a technical effect due to their being capable of controlling hardware such as a printer. It therefore appears that the management information contained in the present claims should be regarded as contributing to the technical character of the search method. The technical effect is the control of the computer along the path leading to the desired data. The path itself, which is determined by the search strategy as reflected by the management information, has technical character for the same reason. Obviously the choice of path will determine the searching characteristics, such as the speed, something which can constitute an additional indication of technicality.' This suggests that when the execution of computer programs on a computer results in controlling an external physical device, the claimed software subject matter is likely to be treated as having a technical character and being patent eligible.

In *T0979/06*, the following comment was made: '[i]n particular, the board has examined whether the subject-matter of claims 11 and 12 has a technical character because the claimed program, when run on a computer, causes a "further technical effect"'. This seems to suggest that having a technical character is the equivalent of having a 'further technical effect'.

In *T1339/11*, the EBA noted that '…the computer program claimed is technical as it serves to control a technical piece of equipment (surgical console). The fact that the computer program is stored on technical means (computer-readable medium) renders it all the more technical'. This was consistent with the T1351/04 decision in which a computer program was patent eligible if, when executing such a computer program, it could interact with or manipulate an external physical entity. The board supported the notion that a computer program stored on technical means was more technical, but this did not necessarily mean it had a technical character.

Despite these efforts, clear guidance for patent eligibility has continued to be absent. This has been especially challenging due to the rapid development of the software industry. In addition, another layer of uncertainty has been added each time an EU member country has interpreted the meaning of 'computer program *as such*' in a different way. For example, despite the shift from the 'technical contribution approach' to the 'technical character approach', and one case, *Symbian*,[118] following this shift, the

[118] *Symbian v Comptroller General*, [2008] EWCA Civ 1066.

UK courts have more or less insisted on using the former approach when assessing whether a program is patentable. The standard of the UK courts has been even higher than the 'contribution approach' because it has required an 'innovative contribution'![119] The EBA has criticised the UK approach for being 'irreconcilable with the EPC'.[120]

3. Software patenting in China

China's Patent Law[121] has no provision specifically provided for software patenting, but only excludes 'rules and methods for mental activities' from patentable subject matters under Article 25.1(2). The Guidelines for Patent Examination ('Guidelines')[122] confirmed that a patent claim directed to 'rules and methods for mental activities' *per se* is unpatentable because it is an excluded subject matter[123] and does not constitute a technical solution under Patent Law Article 2.2.[124] 'Rules and methods for mental activities' are defined as 'human's thinking movements. They originate from human's thinking, and produce abstract results through inference, analysis and judgment, or, via human's thinking movement, produce results by indirectly acting on the nature.'[125] The Guidelines further state that computer programs *per se* are unpatentable because they belong to the 'rules and methods for mental activities'.[126]

However, the Guidelines also state that 'if a claim contains not only matter of rules or methods for mental activities but also technical features, then the claim, viewed as a whole, …shall not be excluded from patentability under Article 25'.[127] Part II, Chapter 9 of the Guidelines is specifically directed to inventions related to computer programs, and distinguishes the concept of 'computer programs *per se*' and the concept of 'invention relating to computer programs'. Computer programs *per se* refers to 'a coded instruction sequence which can be executed by a device capable of information processing, e.g., a computer, so that certain results can be obtained, or a symbolised instruction sequence, or a symbolised statement sequence, which can be transformed automatically into a coded instruction sequence. Computer programs *per se* include

---

[119] *Lantana Ltd. v UK Camptroller general of Patents*, [2003] EWHC 2673. Also see *HTC v Apple* [2013] EWCA Civ 451 and *Aerotel Ltd. v Telco Holdings Ltd.,* [2006] EWCA Civ 1371. The UK Court of Appeals developed a four-step test to assess the 'technical contribution': (1) construe the claim; (2) identify the actual contribution; (3) determine whether the contribution falls solely within the scope of excluded subject matter; and (4) determine whether the contribution is actually technical in nature.

[120] Decision T 154/04 of 15 November 2006, Reasons 13. The board wrote that 'The 'technical effect approach (with the rider)' applied in the Aerotel/Macrossan judgement is irreconcilable with the European Patent Convention also for the further reason that it presupposes that "novel and inventive purely excluded matter does not count as a 'technical contribution" (Aerotel/Macrossan, e.g. paragraph No. 26(2)). This has no basis in the Convention and contravenes conventional patentability criteria (...)'.

[121] The law was adopted in 1984, and has so far been amended three times, in 1992, 2000, and 2008 respectively, see http://www.sipo.gov.cn/zcfg/zcfgflfg/flfgzl/fl_zl/index.htm

[122] The earliest version was released in 2001, available at http://www.cnpat.com/cn_pat/exam_guide_2001.htm. The Guidelines have three official versions, released in 2001, 2006, and 2010 respectively. The Guidelines were also amended in 2008, 2009 (three times), 2013, 2014, and 2017 respectively, available at http://www.sipo.gov.cn/zhfwpt/zlsqzn_pt/zlfssxzjsczn/index.htm

[123] S 3.2 of Cap 1, Part II of the Guidelines (2001); s 4.2 of Cap 1, Part II of Guidelines (2006, 2010).

[124] Article 2.2 prescribes that 'The "Invention" refers to a new technical solution developed for a product, process or the improvement thereof.'

[125] S. 4.2 of Cap 1, Part II of the Guideline (2010).

[126] S 3.2 of Cap 1, Part II of the Guidelines (2001); s 4.2 of Cap 1, Part II of Guidelines (2006, 2010).

[127] *Id.*

source programs and object programs'.[128] On the other hand, an 'invention relating to computer programs' refers to 'solutions for solving the problems of the invention which are wholly or partly based on the process of computer programs and control or process external or internal objects of a computer by the computer executing the programs according to the above-mentioned process'.[129]

Based on such a distinction, patent eligibility for a software claim can be summarised as a two-step inquiry. Step one determines whether a software claim is directed solely to a computer program *per se*. If yes, then the claim is patent ineligible. If not, then the inquiry goes to step two to decide whether the claimed invention constitutes a technical solution. If no technical solution is found, then the claimed invention is patent ineligible. Otherwise, the claimed invention is patent eligible.

Under the Guidelines, a software invention can be considered to constitute a technical solution in the following scenarios:[130]

A. The computer program is executed to control or process external or internal objects, solving one or more technical problems. The claimed invention embodies technical means to achieve technical effects according to the laws of nature.
B. The computer program is executed to process external technical data, which would lead to a 'technical effect regarding data process according to the laws of nature'.
C. The computer program is executed to adjust the components and configurations of the computer system so that its internal performance can be improved.

Part II Software Patenting Dilemma: Comparative Analyses and Proposals for Future Reform

The split of the courts and their inconsistent, even contradictory, rulings on the patentability of software before and after *Alice* in the US, as well as the uncertainty and conflict in defining and ruling on 'computer program as such' and 'computer program per se' have led to confusion in practice. In the US, 'like a tsunami',[131] *Alice* has been used to reject software patents across the country.[132] However, although the 'abstract idea' has been used as the basis for rejection, the courts have not provided any clear definition of 'abstract ideas' and both patent applicants and practitioners have been required to guess what it means.[133] Likewise, in the UK, some patent attorneys have expressed confusion over the rules for software patents and have been unable to

[128] See the Guidelines , Cap. 9, section 1, para. 3.
[129] Sec. 1, Chapter 9, Part II, of the Guidelines.
[130] Sec. 2, Chapter 9, Part II of the Guidelines. The Guidelines use the term 'invention relating to computer programs'.
[131] Eric Guttag, 'The Broken Patent-Eligibility Test of Alice and Mayo: Why We Urgently Need to Return to Principles of Diehr and Chakrabarty'? IPWatchdog, http://www.ipwatchdog.com/2014/09/25/broken-patent-eligibility-test-of-alice-and-mayo/id=51370/.
[132] For example, software patents were invalidated in the following cases: *Planet Bingo, LLC v VKGS LLC* No. 13-1663 (Fed. Cir. 2014); *Buysafe, Inc. v Google, Inc*. 765 F. 3d 1350); *Eclipse IP, LLC v McKinley Equipment Corporation*, No. 15-1023 (Fed. Cir. 2014).
[133] Gene Guinn, 'Ultramercial Patent Claim Invalid as Abstract Ideas,' IPWatchdog, http://www.ipwatchdog.com/2014/11/16/ultramercial-patent-claims-invalid-as-abstract-ideas/id=52194/.

properly advise their clients.[134] China's rejection of software patents based on "technical solution" may also stemmed from the confusion about what is "computer program *per se*". Whether this severe uncertainty will adversely affect innovation in the software industry is unclear, but it may reduce the number of software patents, either through rejection at the patent office or invalidation by the courts, or because there are fewer patent applications as a result of the confusion.

The dilemma in software patenting faced in the three jurisdictions discussed in part I suggests an urgent need to find a worldwide solution to the software patenting problems. To find such a solution, a holistic approach should be adopted to address some fundamental issues such as whether software should be protected by patent. If so, what kind of software can be patented? Specifically, what does 'abstract idea' mean for software, and what does 'computer program as such' or 'computer program per se' mean? If software is to be protected, how long is a proper term for protection?

1. Is Patent Needed for Software Innovation?

There have been two opposite camps on software patents. The camp supporting software patents has argued, *inter alia*, that software developers enjoy a natural property right to their inventions; that they are motivated to invent only when their inventions are justly and fairly rewarded by a property right like a patent, that patents can encourage the disclosure of technologies, and that software patents have some economic benefit.[135] The camp opposing software patents has argued, *inter alia*, that software patents should be abolished because they 'do not promote technological innovation',[136] that they block the entire area of innovation because they are hard to 'invent around', and that it is too costly to license the patented software, conduct patent searches and defend litigation.[137] In between these two camps, there have been some more rational academic discussions focused on identifying the problems and finding ways to fix them.[138]

Some empirical studies have shown that software patents are considered less important to innovation and that many software companies have not sought patent protection for various reasons such as its high cost and the difficulties encountered in

---

[134] A post on the Internet said, 'My business partners and I are filing for a UK patent. The patent is for software and has an application number. … Throughout the entire time, we have been dealing with our patent lawyer we have been unable to get out of him a straight answer on UK software patents. This is likely due to the nature of the invention and the extremely complicated guidelines in the UK. Most people tend to point you to *Aerotel v Telco and Macrossan* as they don't really know the answer themselves'. See Ask Patent, http://patents.stackexchange.com/questions/3183/uk-software-patent.

[135] For general discussion of patent's role in innovation, see Fritz Machlup and Edith Penrose, 'The Patent Controversy in the Nineteenth Century', Journal of Economic History, Vol. X, May 1950, No. 1; and Yahong Li, Intellectual Property and Innovation: A Case Study of High-Tech Industries in China, 101 Oregon Review of International Law, Vol. 13, X, 2011; Also see Benjamin Coriat, 'Establishing a New Intellectual Property Rights Regime in the United States: Origins, Content and Problems', Research Policy 31 (8-9): 1491-1507.

[136] Amici Brief of Google, et al. for the case of *Bilski v David Kappos*, at 2-3, http://www.americanbar.org/content/dam/aba/publishing/preview/publiced_preview_briefs_pdfs_09_10_08_964_RespondentAmCu7FinancialCorps.authcheckdam.pdf (last visited 4 September 2015).

[137] Pieter Hingtjens, 'In defense of software patents', Free Software Magazine, Tue. 2007-11-27, http://www.freesoftwaremagazine.com/articles/in_defense_of_software_patents.

[138] For example, Eric Goldman provided some interesting ways to fix software patents. See Eric Goldman (2013), 'Fixing Software Patents', Santa Clara University School of Law, Legal Studies Research Paper No. 01-13, https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2199180.

acquiring and enforcing patents.[139] In reality however, companies like Google, while openly condemning software patents,[140] have strived to build robust software patent portfolios. For example, Google acquired 2,566 US patents in 2014 alone, ranking eighth in patent acquisition from the USPTO.[141] This phenomenon indicates the value of software patents and the practical need for them. The history of US software patenting discussed above[142] also demonstrates that software patents have been the natural result of technological advancement, particularly Internet technologies, and have not been something imposed by lawmakers or courts. Given that computer programs have become more sophisticated and used to facilitate and manage complex industrial or business processes that generate unprecedented business value, programmers and software companies would naturally seek the best way to protect their programs. Patents have the advantage of preventing others from using the patented processes or products without authorisation once the invention has filed for patent protection. This is superior to the traditional modes of trade secrets or know-how protection because the latter cannot prevent a competitor from independently developing the same program. Because software patenting has existed for more than four decades and is still popularly accepted among software developers, it would be very difficult to make a conclusive case for abolishing it.[143]

2. What Computer Program is '*as such*', '*per se*" or an 'Abstract Idea' thus Patent Ineligible?

On the premise that patent is necessary for software innovation, this section explores software patent eligibility issue, that is, what makes a software eligible or ineligible for patent protection? In Part I, we have seen that all three jurisdictions, the US, Europe and China, have adopted some vague terms to prevent certain software from being patentable. The USPTO uses 'abstract idea', the EPO uses 'computer program as such', and the CNIPA uses 'computer program per se', which more or less mean the same thing, that is, computer programs that are pure coded instructions and have no practical application. To make these 'pure coded instructions' patent eligible, all three jurisdictions require certain kind of 'practical elements' of the software such as 'practical application' in the US, 'technical contribution', 'technical effect', and 'technical character' in Europe, and 'technical solution' in China. Comparatively

---

[139] Stuart J. H. Graham, Robert P. Merges, Pam Samuelson, and Ted Sichelman, 'High Technology Entrepreneurs and the Patent System: Results of the 2008 Berkeley Patent Survey,' 1255 Berkeley Technology Law Journal, Vol. 24:4 [2009]. Specifically, the statistics in the survey showed that about two thirds of the 700 software companies surveyed neither had nor were seeking patents for their inventions. They rated patents as the least important tools among seven options to attain a competitive advantage in the marketplace. Even software startups viewed patents as an insignificant incentive to invest in innovation. The cost of obtaining and enforcing a patent was the number one factor affecting software firms' decision not to seek a patent. More than 40% of the firms surveyed cited the un-patentability of the invention as a factor.

[140] See note 141 above.

[141] Gene Guinn, 'Google collects patents while lobbying against them', IPWatchdog, http://www.ipwatchdog.com/2015/05/03/google-collects-patents-while-lobbying-against-them/id=57302/ .

[142] See [Source], *supra,* section 3 (1).

[143] This expression is borrowed from Fritz Machlup's conclusion on patent systems in general, in which he stated, 'since we have had a patent system for a long time, it would be irresponsible, on the basis of our present knowledge, to recommend abolishing it'. Fritz Machlup (1958), An Economic Review of the Patent System, Study No. 15 of Comm. On Judiciary Subcomm. On Patents, Trademarks, and Copyright, 85th Cong., 2d Sess.

speaking, eligibility and patentability discourse in the US and Europe is more sophisticated than in China because of the abundant cases decided in these two jurisdictions.[144]

To help examiners and practitioners better understand what 'abstract idea' and 'more than abstract idea' mean in a software claim, the USPTO issued 'Eligibility Quick Reference Sheet Identifying Abstract Ideas' in July 2015, and then updated in July 2018, identifying eight hypothetical examples[145] which has been repealed and replaced by the USPTO 2019 Revised Patent Subject Matter Eligibility Guidance ("Guidance") on 7 January 2019,[146] which clarifies the patent examination process under 35 U.S.C. §101. Likewise, EPO also updated its Guidelines for Examination ('Guidelines') in November 2018, making changes primarily on patent eligibility of computer-implemented inventions (CII). China amended its Examination Guidelines in February 2017 as well to further distinguish 'computer program per se' and 'computer implemented inventions'.

2.1 USPTO '2019 Revised Patent Subject Matter Eligibility Guidance'

Under the Guidance, two steps are required to decide the eligibility: step one decides whether the claim belongs to a statutory category of patentable subject matter; step two includes two sub-steps: 2A and 2B. Step 2A decides (1) whether the claim recites a judicial exception; and (2) whether a recited judicial exception is integrated into a practical application. If the answer to 2A(1) is "yes" and 2A(2) is "no", then step 2B is required to decide whether the claim merely recites well-understood, routine, conventional activity.[147]

In addition to the clarifications on the procedure, the Guidance makes the following groups of "abstract ideas" based on the courts' decisions:[148]

(a) Mathematical concepts – mathematical relationships, mathematical formulas or equations, mathematical calculations;

(b) Certain methods of organizing human activity – fundamental economic principles or practices (including hedging, insurance, mitigating risk); commercial or legal interactions (including agreements in the form of contracts; legal obligations; advertising, marketing or sales activities or behaviours; business relations); managing personal behaviour or relationships or interactions between people (including social activities, teaching, and following rules or instructions);

(c) Mental processes – concepts performed in the human mind (including an observation, evaluation, judgment, opinion).

In the "the rare circumstance", an examiner can treat a claim limitation not falling within the above groupings as reciting an abstract idea. In such cases, the examiner should evaluate whether the claim as a whole integrates the recited "tentative abstract idea" into a practical application. If so, the claim is not directed to a judicial

---

[144] See Part I, section 1 and 2.

[145] The Reference has been repealed by the revised guidance in January 2019, and the webpage http://www.uspto.gov/patents/law/exam/abstract_idea_examples.pdf does not exist anymore. The Revised Guidance can be found at: Department of Commerce, USPTO '2019 Revised Patent Subject Matter Eligibility Guidance',[Docket No PTO-P-2018-0053], https://www.federalregister.gov/documents/2019/01/07/2018-28282/2019-revised-patent-subject-matter-eligibility-guidance

[146] Notice of Federal Register, Vol. 84, No. 4, see https://www.govinfo.gov/content/pkg/FR-2019-01-07/pdf/2018-28282.pdf

[147] See note 151 above, at 5.

[148] Ibid. at 9-11.

exception and thus eligible; otherwise, the examiner shall evaluate additional elements to decide whether there is an inventive concept under step 2B.[149]

The USPTO also announced guidance on the application of 35 U.S.C. § 112 to computer-implemented inventions, which describes proper application of means-plus-function principles under § 112(f), definiteness under § 112(b), and written description and enablement under § 112(a).[150] This will not be the focus of this article due the space limit.

2.2 The EPO 2018 Guidelines Revision

Under the Guidelines, only the computer programs that have 'technical character' are eligible for patent protection.[151] Here, the 'technical character' refers to a 'further technical effect' that goes beyond the 'normal' physical interactions between software and the hardware.[152] The 'further technical effect' has to be able to control a computer's technical process or internal functioning or its interfaces.[153] It cannot be merely claimed as such, but has to be 'reflected in claimed features that cause a further technical effect'.[154] Some examples of 'further technical effects' are provided such as methods 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; implementing security measures for protecting boot integrity or countermeasures against power analysis attacks; processor load balancing or memory allocation; building runtime objects from development objects, regenerating only those runtime objects resulting from modified development objects.[155] Only the computer programs having the above 'technical character' (by looking at whether there is any further technical effects) are eligible for patent protection, and are subject to the next test of novelty and inventive step.[156]

2.3 The CNIPA 2017 Amended Guidelines

On 1 April 2017, a new amendment was introduced into the Patent Examination Guidelines of China National IP Administration (CNIPA).[157] The amendment clarified some wordings so that the differences between 'computer program *per se*' and 'inventions relating to computer program' have become more distinct. For example, in section 2 of Part II, Cap 9, '*per se*' was added to 'computer programs recorded in mediums' and 'recorded program' to clarify that the computer program 'per se', even

---

[149] Ibid. at 25.
[150] https://s3.amazonaws.com/public-inspection.federalregister.gov/2018-28283.pdf
[151] EPO Examination Guidelines 2018, G-II, 3.6, para. 1.
[152] More specifically, 'A "further technical effect" is a technical effect going beyond the "normal" physical interactions between the program (software) and the computer (hardware) on which it is run. The normal physical effects of the execution of a program, e.g. the circulation of electrical currents in the computer, are not in themselves sufficient to confer technical character to a computer program (T 1173/97 and G 3/08).' Ibid. para. 2.
[153] Ibid. para. 3.
[154] Ibid. para. 5
[155] Ibid. G-II, 3.6.1
[156] Ibid. G-II, 3.6, para 6.
[157] This amendment was publicly consulted under the 'Notice on Public Consultation on 'Draft Amendment to Guidelines for Patent Examination' (Draft for Comment)'issued by then State Intellectual Property Office (SIPO) (after 28 August 2018 SIPO retitled China National Intellectual Property Administration or CNIPA) on 27 October 2016. See http://www.sipo.gov.cn/gztz/1099153.htm; Also see Decision on Amendment to the 'Guidelines for Patent Examination' (2017) (Order No. 74). http://www.sipo.gov.cn/zcfg/zcfgflfg/flfgzl/zlbmgz/1020135.htm

it is recorded in mediums such as tapes, discs, is not patentable. The amendment also allowed more ways to formulate patent claims for software inventions such as 'a computer program product', 'a machine-readable medium', and 'an apparatus comprising a processor configured to execute instructions on a computer-readable medium to perform steps of ...'[158] Many commentators have regarded the amendment as being pro-software patent.[159] However, it added nothing substantively new to patent eligibility for software patents. Also absent have been judicial interpretations and court decisions.

2.4 Summary and comments

From the revised USPTO Guidance and the Guidelines of the EPO and the CNIPA, we can see that all three jurisdictions have striven to distinguish two types of computer programs: the one that is "*as such*', '*per se*' or 'abstract idea', therefore is patent ineligible; and the other with something technically more to be patent eligible.

Comparatively speaking, the US construct on patent eligibility is more complicated but may allow more computer programs that are seemingly 'abstract ideas' filtered into patent protection. Under the USPTO 2019 Guidance, the claims directing to 'abstract idea' that is not integrated into a practical application can be subject to further test of whether the claim not merely recites 'well-understood, routine, conventional activity', or whether there is anything 'significantly more' than the recited 'abstract idea', or presents an 'inventive step'.[160] This may reverse the trend set by *Alice* and give "abstract idea'-type of programs more chance to get patent protection. The EPO and the CNIPA constructs on software patent eligibility are similar in that both exclude '*as such*' or '*per se*' computer programs at the step-one (eligibility) inquiry without giving them a chance such as the step '2A(2)' in the US.[161] Only those programs that have 'technical characters' (in the EPO context) or 'technical solutions' (in the CNIPA context) can advance to step-two (patentability) inquiry. Arguably, EPO's 'further technical effect' inquiry resembles USPTO's 2A(2) step, but it does not clearly say that the inquiry is carried out after the program is deemed 'as such'.

Another problem worthy of discussion is the lack of a generalised definition of 'abstract idea' in US Guidance and 'computer program as such' in EPO Guidelines, although there is one for 'computer program per se' in CNIPA Guidelines.[162] The grouping of the 'abstract ideas' provided by the USPTO[163] is a positive step,[164] but may

---

[158] Richard Huang, 'China will lift curbs on software patents as of April 1, 2017 -- SIPO revised the Patent Examination Guidelines', LinkedIn (1 March 2017), see https://www.linkedin.com/pulse/china-lift-curbs-software-patents-april-1-2017-sipo-richard-huang?trk=v-feed&lipi=urn%3Ali%3Apage%3Ad_flagship3_feed%3BENVekQ%2BCg6VsppSOhezdTw%3D%3D

[159] See e.g., Steve Brachmann, 'China relaxing barriers to software, business method patents with revised patent guidelines', IPWatchdog (3 March 2017), http://www.ipwatchdog.com/2017/03/03/china-relaxing-barriers-software-business-method-patents/id=79017/

[160] Department of Commerce, USPTO '2019 Revised Patent Subject Matter Eligibility Guidance',[Docket No PTO-P-2018-0053], at 22-23. See https://s3.amazonaws.com/public-inspection.federalregister.gov/2018-28282.pdf

[161] See Part II, 2.1 above.

[162] See Part I, section 3 of this article and note 132 above.

[163] See Part II, section 2.1 of this article and note 150 above.

[164] The 'grouping' has been hauled as marking 'a positive step forward for patent owners, both in terms of clarity of the examination procedure and opportunities to argue for eligibility of patent claims'. See S. James Boumil, III, Joseph A. Capraro Jr. and Patrick J. Niedermeier, "United States: USPTO Announces 2019 Revised Patent Subject Matter Eligibility Guidance', Mondaq, 21 January 2019,

not be helpful when claims go beyond the situations described in the 'groupings'. The 'rare circumstance' provided in the Guidance[165] sounds abstract itself and may cause more confusion than clarity. Admittedly, it is difficult, if not impossible, to define 'abstract idea' or computer program 'as such' or 'per se' because software is very complicated and varies from one case to another. Nonetheless, it is necessary to define the terms because '[w]ithout a definition for the term *abstract idea* rulings have been nothing short of subjective; some would even say arbitrary and capricious'.[166] A key but undefined term would always be a source of confusion in deciding patent eligibility. Notwithstanding the difficulty, this article argues that it should be feasible to have a general definition because there have been rich laws and practices in the US and Europe, and China's definition of 'computer program per se' may shed some light.[167] 'Abstract idea' is defined in a dictionary as 'thoughts in the mind and are not related to physical objects or real events and actions'.[168] The USPTO's 'grouping of abstract ideas' of mathematical concepts, certain methods of organizing human activity, and mental processes[169] has generally reflected the features in the dictionary and can be used as a basis of the definition. However, the definition of 'abstract idea' under USPTO Guidance is not only applicable to software but also to all other subject matters. It cannot be applied to define computer program 'as such' or 'per se'. Therefore, there could be three different solutions to this problem: (1) Europe and China, as well as other countries,[170] can adopt a US-like 'abstract idea' groupings to cover all subject matters including computer program. (2) European countries and other countries adopt a separate definition for computer program 'as such', similar to the one in the CNIPA's Guidelines.[171] Or (3) This article suggest a general definition of 'abstract idea' for all jurisdictions covering all types of patent subject matters as follows: 'Abstract ideas that render subject matter unpatentable are the thoughts of mind without practical application to any physical objects or real actions and any technical results, including but not limited to mathematical concepts, methods of organizing human activity and mental processes'.

As mentioned above, the step 2A(2) under the revised USPTO Guidance provides the claim directing 'abstract idea' one more chance to be evaluated for patent eligibility with an assessment of its 'inventive concept'. This is plausible because it

---

http://www.mondaq.com/unitedstates/x/773146/Patent/USPTO+Announces+2019+Revised+Patent+Subject+Matter+Eligibility+Guidance.

[165] See Part II, section 2.1 of this article and note 233 above.

[166] Gene Quinn, 'Revised Patent Eligibility Guidance Effectively Defines What is an Abstract Idea', IPWatchDog, 4 January 2019, https://www.ipwatchdog.com/2019/01/04/patent-eligibility-guidance-abstract-idea/id=104754/

[167] See Part I, section 3 of this article and note 152 above.

[168] MACMILLAN English Dictionary, 2007 edition, at 6.

[169] See note 232 above. Some commentators also coincided with this view. See Gene Quinn, 'Revised Patent Eligibility Guidance Effectively Defines What is an Abstract Idea', IPWatchDog, 4 January 2019, https://www.ipwatchdog.com/2019/01/04/patent-eligibility-guidance-abstract-idea/id=104754/

[170] For example, in Japan, software related inventions are patentable only when there is 'a creation of technical ideas utilizing a law of nature'. See "Examination Guidelines for Patent and Utility Model in Japan (REQUIREMENTS FOR PATENTABILITY)" (PDF). jpo.go.jp. pp. 1–3. In South Africa, a program for a computer as such is excluded from patent protection under section 25(2) of the Patents Act. *Patents Act, No. 57 of 1978, as amended Section 25(3)(PDF), 26 April 1978.*

[171] 'Computer programs per se said in this Chapter mean a coded instruction sequence which can be executed by a device capable of information processing, e.g., a computer, so that certain results can be obtained, or a symbolised instruction sequence, or a symbolised statement sequence, which can be transformed automatically into a coded instruction sequence'. See CNIPA Guidelines, Part II, Cap 9, sec. 1, para. 3, note 152 above.

does not stop at the 'abstract idea' but rather goes beyond it to see whether those seemingly 'abstract ideas' are still worthy of patent protection. However, this not only has not completely solved the problem of overlapping between eligibility and patentability assessment, as indicated from the cases in both the US[172] and Europe,[173] but also may blur these two steps even more. It is true that the revised US Guidance asks the examiners to only consider in step 2A(2) whether 'abstract idea' has been integrated into a 'practical application', and leave the analysis of 'inventive concept' to step 2B.[174] However, it is unclear what 'practical application' actually means, which may cause confusion in practice. The overlapping issue is also evidenced in the process of patent examination in China. In practice, Chinese examiners normally skip the first-step inquiry on the issue of eligibility and directly proceed to the second-step inquiry on technical solution. According to Liu's study,[175] among 681 re-examination proceedings for software business method applications, 74.6% (or 508 cases) of the rejections were based on Article 2(2) (technical solution test), 19.53% based on Article 22(3) (inventiveness test), and only 1.17% based on Articles 25 (eligibility).[176] Among the 508 cases rejected under Article 2(2), only 7.5% cited prior art references.[177] These figures indicate that, under most circumstances, the examiners would reject software patent applications based on their subjective opinions about whether the inventions constitute technical solutions, without considering eligibility and conducting prior art search. This practice is in contrast to the EPO and the USPTO. For example, among 287 EPO examined applications, 85% evaluated eligibility and inventive steps based on prior art search.[178] Among 360 USPTO examined applications, 88% evaluated eligibility and conducted prior art searches.[179] The findings also show the conflicting attitudes towards software patenting between the statutory framework and the practices of the Patent Office in China. While the Guidelines have been amended to favour software patenting, the Patent Office has readily rejected software claims for lack of "technical solution".

In addition to the definition of 'abstract idea', the scope of patent claims should be noted. Professor Mark Lemley pointed out that 'functional claiming of software inventions is arguably responsible for most of the well-recognised problems with software patents today'.[180] By claiming 'function' rather than a specific means of implementing that function, a patent can cover any possible means to a goal that 'naturally lead[s] to patent thickets'.[181] He proposed that claims for the means to implement a function be allowed, but not the function itself, so that any developer could come up with the same function in different ways.[182] This article argues that, in defining 'abstract idea', the lawmakers or judges should also take the 'means versus function' into account because 'function' in software is relevant if not equivalent to 'idea' in a

---

[172] See the discussion in Part I, section 1.3.3 of this article about the overlap between§101 and§102, 103.
[173] See the discussion in Part I, section 2.3, note 131 above.
[174] USPTO 2019 Revised Patent Subject Matter Eligibility Guidance, III-A-2, at 19.
[175] Ming Liu, 'Examination Methods for Patent Applications Involving Business Methods' (23 June 2014), available at http://www.sipo-reexam.gov.cn/zxzx/mtbd/20516.htm (28 Apr 2018).
[176] *Ibid*.
[177] *Ibid.*
[178] *Ibid.*
[179] *Ibid.*
[180] Mark Lemley, 'Let's Go Back to Patenting the 'Solution,' Not the 'Problem',' Wired, http://www.wired.com/2012/10/mark-lemley-functional-claiming/.
[181] *Ibid.*
[182] *Ibid.*

general sense. This issue has been addressed in the 2010 revised Guidance.[183] The Guidance states that 'A claim limitation is presumed to invoke 35 U.S.C. § 112(f) when it explicitly uses the term "means" and includes functional language'.[184] The guidance also states: 'for a computer-implemented … claim limitation, the specification must disclose an algorithm for performing the claimed specific computer function, or else the claim is indefinite'.[185] It was reported that the USPTO has recently issued more section 112 rejections against computer-implemented invention claims, even without 'means-plus-function' language.[186]

3. How long software should be protected?

The debate on whether 20-year protection for a software patent is too long has never gone away because the product lifecycle for software can be as short as two and a half years.[187] Richard Posner proposed a shorter patent term in industries whose innovation costs are low and life-cycles are short, and that have market competitive advantages.[188] Eric Goldman pointed out that software has a shorter innovation cycle and lower capital requirements. Therefore, according to him, software development would benefit from much shorter patent lives.[189]

Changing the length of patent terms for a particular field of technology is a sensitive matter because (1) the 20 year term for protection of software patents has been in practice for over three decades and it would be very hard to change the path-dependency; (2) it would require Congressional action which involves public consultation and complicated proceedings; and (3) it may invoke the question of whether a shorter term for software patents would violate the 'non-discrimination' provision in the TRIPS Agreement,[190] and the provision on the patent term which requires patent to be protected for at least 20 years from the filing date.[191] Nevertheless, this article argues that the cost of retaining a long outdated and an obviously unsuitable and problematic term far outweighs the difficulty of changing it. As to TRIPS 'non-discrimination' provision, it applies to the situation in which a specific field of technology is prevented from obtaining protection or where a given invention is discriminated against based on where it comes from. It does not specifically apply to how long the protection is provided to the technologies. In fact, providing less than 20

---

[183] See USPTO, Examining Computer-Implemented Functional Claim Limitations for Compliance with 35 U.S.C. 112 [Docket No. PTO-P-2018-0059.
[184] Ibid. at 6.
[185] Ibid. at 10.
[186] Jim Singer, 'USPTO updates patent examination guidelines for computer-implemented inventions: Less abstract ideas, more specific algorithms', IP Spotlight, 7 January 2019, https://ipspotlight.com/2019/01/07/uspto-updates-patent-examination-guidelines-for-computer-implemented-inventions-less-abstract-ideas-more-specific-algorithms/
[187] USPTO, 'Performance and Accountability Report, Fiscal Year 2015', https://www.uspto.gov/sites/default/files/documents/USPTOFY15PAR.pdf
[188] Richard Posner, 'Why There Are Too Many Patents in America,' The Atlantic, 12 July 2012, http://www.theatlantic.com/business/archive/2012/07/why-there-are-too-many-patents-in-america/259725/.
[189] Eric Goldman, 'The Problems with Software Patents', Forbes/Tech, 29 Nov. 2012, http://www.forbes.com/sites/ericgoldman/2012/11/28/the-problems-with-software-patents/.
[190] TRIPS Article 27.1 requires that patents 'be available and patent rights enjoyable without discrimination as to the place of invention, the field of technology and whether products are imported or locally produced'.
[191] TRIPS Article 33 provides, 'The term of protection available shall not end before the expiration of a period of twenty years counted from the filing date'.

years patent protection, e.g., 19.5 years protection for pharmaceuticals due to stockpiling exception under the Canadian Patent Law section 55.2(2)[192] was decided by the WTO dispute settlement body (DSB) to be "substantial curtailment of the exclusionary rights" violating TRIPS Article 28.1 about exclusive rights, not Article 27.1 concerning non-discrimination.[193] In paragraph 7.91 of the DSB decision, the panel ruled that the TRIPS non-discrimination provision is applicable to exceptions granted under Article 30 and Article 31, but it does not mention whether Article 27.1 might also be applicable to Article 33 patent term provision.[194] It is arguable that, since Article 33 explicitly requires at least 20-year patent protection, any term shorter than 20 years might be suspected to be discrimination. Interestingly, however, different categories of technologies with shorter protection periods have already existed and have never been challenged for violating TRIPS' non-discrimination clause. These include, for example, short-term patents or utility models and design patents, as discussed below.

Then, how long should software be protected? Brian Love found that long terms for software patent protection have caused patent trolling in this field because '17 to 20 years down the road is when troll-owned, software patents are most often asserted…'.[195] He proposed that the patent term should be shortened by three years or even longer.[196] Defend Innovation, a project of the Electronic Frontier Foundation (FEF), has called for shortening the term from 20 years to 5 years, at least in the United States.[197] I would suggest 10 years, which is shorter than Professor Love's 17-years and longer than EFF's 5-years. The 10-year period is proposed based on the following considerations. First, many countries have provided 7-10 years of patent protection to inventions that have received less stringent examinations, under the regimes of utility models, short-term patents, petty patents, or innovation patents. Therefore, adopting a similar length of protection for software would not deviate too much from standard practice, and could be easier to implement because there are existing formats to adopt. Second, although 'a particular software innovation has a two-year commercial lifecycle', [198] many important ones may have longer lifecycles that need longer protection. Third, a typical patent examination period is between two to three years. Therefore, it is reasonable to give the innovators a seven to eight years of monopoly after the patent is granted. In this way, they can recoup their investment if their inventions are truly worthy of such protection. Last but not the least, it is notoriously difficult to enforce a patent, e.g., 4-7 years and $5M -10M+ to succeed in a patent lawsuit, and it is even harder to enforce software patents due to the trend of anti-software patents.[199] For example, *Alice* case took seven years (2007 to 2014) to conclude. 10 years of duration may ensure that the patent is still valuable and worthy

---

[192] This section has been repealed since 2001.
[193] See 'DS114: Canada – Patent Protection of Pharmaceutical Products', WT/DS114/R, 2000, https://www.wto.org/english/tratop_e/dispu_e/cases_e/ds114_e.htm.
[194] *Ibid*, at 170, para. 7.91.
[195] See Brian Love, 'Could a Patent Term Reduction Solve the Software Patent Problem'? http://www.theatlantic.com/business/archive/2012/07/why-there-are-too-many-patents-in-america/259725/.
[196] Brian Love, 'An Empirical Study of Patent Litigation Timing: Could a Patent Term Reduction Decimate Trolls Without Harming Innovation'? 161 University of Pennsylvania Law Review 1309, at 1350.
[197] Richard Stallman, 'Shorten the Patent Term,' Defend Innovation, 21 June 2012, https://defendinnovation.org/proposal/shorten-patent-term.html
[198] Eric Goldman, 'The Problems with Software Patents,' Forbes/Tech, 29 Nov. 2012, http://www.forbes.com/sites/ericgoldman/2012/11/28/the-problems-with-software-patents/.
[199] Roy Schestowitz, 'Patents in the US Are Not Hard to Enforce, Software Patents Are', Techrights, 25 February 2018, http://techrights.org/2018/02/25/software-patents-in-the-us/

of protection when the infringement takes place and the lawsuit is being filed. This proposal, on one hand, takes account of the uniqueness of the software industry and that software inventions are generally based on existing products and are improvements to the same.[200] Modest protection would provide innovators with the necessary incentives without sacrificing the development of the entire software industry. On the other hand, the proposal recognises the cost of applying and prosecuting patent applications. A period of protection that is too short would end with the cost of seeking a patent extending beyond the benefit, thereby discouraging innovators from pursuing patents for their software inventions. As a result, the patent system would play no role in facilitating software innovations.

4. Solutions to the Software Patenting Dilemma

The above discussions demonstrate that software patenting has become one of the most, if not the most, debated issue in the patent field worldwide. Countries as well as national courts have been seriously divided over the issues such as eligibility of software patents. *Alice* represented the climax of such division and the hopelessness of seeking any consensus, even among the once fairly unified the CAFC. As discussed above, *Alice* opened the floodgates to reject software patents. However, post-Alice development indicates that rejecting software patents does not solve all the problems over the long run, as it may kill true innovation in the software industry if the innovators do not receive the protection they deserve.[201] The USPTO, EPO and CNIPA had quickly reacted to the doubt cast in software industry by revising their patent examination guidance or guidelines to further clarify some ambiguous and misunderstood concepts such as 'abstract idea', 'further technical effect', 'computer program per se' respectively under the three guidelines. But these examination guidelines are patent offices' internal working manuals. They are not laws and do not have binding effect on the courts. To solve software patent dilemma faced by software industry, law practitioners, and courts around the world, a more holistic international approach which goes beyond mere defining 'abstract idea' is needed.

4.1 Solutions at the national law level

In 2005, Burk and Lemley endeavoured to design an 'optimal software patent' policy that could 'maximise the patent incentive effect and minimise the burdens placed on future innovation. They suggested tailoring the scope of the patent to the characteristics of each technological sector affected'.[202] According to them, software patents should have a narrower scope and be easier to obtain based on the following characteristics of software innovations: software is a cumulative, on-going or incremental innovation with a low cost and a short product cycle. Conversely, computer programs are built on prior codes and require interaction to achieve interoperability. In addition, they depend

---

[200] Dan L. Burk and Mark A. Lemley, Policy Levers in Patent Law, (2003) 89(7) Virginia Law Review 1575-1696, at 1620.

[201] Some patent attorneys have questioned whether the trend of rejecting software patents brought by *Alice* would affect innovation in the entire industry. See Gene Guinn, 'Is there a future for software patent in an age of software innovation'? IPWatchdog, http://www.ipwatchdog.com/2015/05/26/future-for-software-patents-age-of-software-innovation/id=58088/.

[202] Dan L. Burk and Mark A. Lemley, 'Designing Optimal Software Patents', Stanford Public Law Working Paper No. 108, and Minnesota Legal Studies Research Paper No. 05-11, abstract.

on reverse engineering to produce new software.[203] As for the scope of software patents, Lemley suggested they be limited to a 'specific method (or means)' of implementing function only, and not extend to the 'function'.[204] This suggestion has been adopted into the USPTO revised Guidance.[205]

Many scholars have proposed other solutions.[206] For example, there was John Duffy's approach to deciding obviousness for software whose multiple inventors achieved the same solution around the same time. James Bessen and Brian Love proposed that maintenance fees for patents (including software patents) be increased, particularly as the patent ages and is being asserted, so that some weak patents could lapse earlier. Jennifer Urban and Jason Schultz proposed the Defensive Patent License (DPL) under which patent owners agree to license their patents royalty-free to everyone who has opted into the DPL, and everyone in the DPL agrees not to sue each other for patent infringement.

One solution proposed by Eric Guttag was particularly interesting because it directly related to the 'abstract idea' problem discussed above. In this proposal, Guttag suggested a 'safe harbour' be made for software patent-eligibility, under which 'electronic/computer/machine implementation should be viewed as taking the claimed business (or system) out of being merely an 'abstract idea,' and into a 'safe harbour' for patent-eligibility'.[207] This proposal could potentially alleviate the difficulty over defining 'abstract idea' for three reasons: (1) This proposal is closer to the test of 'machine-or-transformation' adopted by the US courts in the 1980s, when software patenting was just taking off into a relatively cautious and healthy development environment. (2) The proposal is similar to the 'technical character' criterion used by the EPO and European courts, emphasising the means used to implement the solution to the problem. (3) The concept of 'safe harbour' may provide a way out for 'abstract idea'. That is, if it is difficult to arrive at a clear definition of what an 'abstract idea' is, the patent examiners and judges could put the apparently non-abstract-ideas (e.g., electronic/computer/machine implementations) into a safe harbour and focus only on the more questionable claims.

4.2 Solution at the international level

The unprecedented division over software patents and the wave of rejection after *Alice,* in addition to the conflicting rulings in Europe and the contradictions between law and practice in China suggests that a more holistic solution for software patenting is needed in the US and other countries. Software patents concern mostly computer networks, and computer networks are borderless. Thus, a more unified solution at the international level would help to end the battle over software patent eligibility and create a healthier legal regime for the software industry.

The international effort for software patent legislation should be led by the World Intellectual Property Organization (WIPO), not the World Trade Organization (WTO) because the WTO's TRIPS has no provision specifically dealing with software

---

[203] *Ibid.*
[204] Mark Lemley, note 279 above.
[205] See notes 282-285 above.
[206] See Eric Goldman, 'The Problems with Software Patents', Forbes/Tech, 29 Nov. 2012, http://www.forbes.com/sites/ericgoldman/2012/11/28/the-problems-with-software-patents/.
[207] Eric Guttag, 'CLS Bank International: A Fractured Landscape of Patent Eligibility for Business Methods and Systems', IPWatchdog, 12 July 2012, http://www.ipwatchdog.com/2012/07/12/cls-bank-international-a-fractured-landscape-of-patent-eligibility-for-business-methods-and-systems/id=26342/.

patenting.[208] Further, it is unlikely that it would deal with such a technical problem in the near future. In addition, the WTO is a trade organisation that has been suspected of bias towards under-developed countries by more industrially advanced countries, particularly when formulating IP policies on cutting-edge technologies such as software. WIPO, however, has been perceived as an organisation striving for international cooperation. Additionally, it is an organisation that specialises in IP and it has expertise in formulating IP policies for more sophisticated technologies. One of WIPO's missions has been to 'shape international IP rules for a changing world'.[209] In the past few decades, WIPO had successfully led the drafting and implementation of many international IP conventions and treaties such as the WIPO Internet Treaties 1996[210] and the Beijing Treaty on Audio-visual Performances 2012. It has also drafted the 'model provision for unfair competition' and the 'joint recommendation for the protection of marks on internet', among others.

It is worth noting that WIPO-led several initiatives to draft the Model Provision on the Protection of Computer Software in 1978 (Model Provisions);[211] and the Draft Treaty for the Protection of Computer Software in 1983 (Draft Treaty).[212] Both the Model Provisions and the Draft Treaty proposed a *sui generis* system for software protection, not patent protection. The Draft Treaty received insufficient support from the member states and was not adopted. The Model Provisions have not been followed by national legislation.

Given the present uncertainty surrounding software patenting, it is time for WIPO to consider drafting more up-to-date 'model provisions' or WIPO-Paris Union 'joint recommendations' on patent protection for computer software, particularly on the issues of patentability and the eligibility of software. These should provide guidance on the kind of software inventions that are eligible for patent protection, and specifically, what an 'abstract idea' or a 'computer program as such' (or mathematic algorithms) is, and how long such a software patent should be protected. It is hoped that if these issues are carefully studied and clearly explained, the model provisions or joint recommendations would be followed by national legislation. If that happens, a more sophisticated international treaty on software patenting could be adopted so that the world connected by Internet would have more uniform laws governing the protection of software innovations. In the meantime, it should be acknowledged that this is not by any means an easy task as the WIPO has been plagued with many daunting negotiations such as the WIPO Broadcast Treaty, the negotiations on IP, Genetic Resources and Traditional Knowledges, and WIPO Treaty on Design Law, all of which have been dragging for years by the fiercely conflicting agenda from developed and developing/least developed countries.

Conclusion

[208] Under TRIPS Article 27 (3), computer program (or software) is not in the list of exclusions. Therefore, it has been interpreted that computer program are patentable under TRIPS.
[209] WIPO webpage at http://www.wipo.int/portal/en/index.html
[210] WIPO Internet Treaties consist of two treaties: the WIPO Copyright Treaty (WCT) and the WIPO Performances and Phonogram Treaty (WPPT) adopted in 1996.
[211] For a full introduction and discussion on this model provision, see Yoshiyuki Miyashita, 'International Protection of Computer Software', 11 Computer/Law Journal, 41 (1991).
[212] See http://www.valimaki.com/org/docs/wipo_draft_treaty_1983.pdf.

The unprecedented division as to patent eligibility based on determination of 'abstract idea' among the courts and judges in *Alice v. CLS Bank* and post *Alice* development, as well as the uncertainty as to the concept of "computer program *as such*' in Europe and the conflict in law and practice in deciding "computer program *per se*" in China show that software patenting has come to a crossroads and its future direction is uncertain. Since *Alice*, an alarming number of software patents have been rejected or invalidated in the United States. This has been seen by some as a setback and has been called a 'tsunami'. Although an outright abolition of software patenting is unlikely to occur in the US or elsewhere in the near future, the current state of rejection and invalidation could have a chilling effect on the software industry. Software developers are in a double bind. On the one hand, their incentive to write program with true value could be reduced if they fear their inventions will not receive patent protection. Thus, they may discontinue their efforts. On the other hand, if they continue to develop and apply for patents their programs could be rejected, not because they lack technical value but because of the trend to reject applications based on the ambiguous concept of 'abstract idea'. Software developers and their patent attorneys could also face tremendous difficulties trying to ascertain what 'abstract ideas' are and how to avoid them when writing program and drafting patent claims.

The recent revisions to patent examination guidance or guidelines, e.g., as to some ambiguous concepts, particularly the USPTO's 'grouping' the 'abstract ideas', providing examples of 'further technical effect' by the EPO, and distinguishing 'computer program *per se*' and 'computer-implemented inventions' by the CNIPA, sent out a clear signal that software is to be treated the same as any other invention and will not be out-rightly rejected or invalidated as an 'abstract idea'. Although helpful in patent examinations, the role of the above efforts are limited to the specified scenarios. A more general definition, or a summary of the general features of an 'abstract idea', or computer program '*as such*' or '*per se*' for computer software is needed to solve the problem from the root. Alternatively, a 'safe harbour' for abstract idea or '*as such*' or '*per se*' computer programs could avoid having to decide what is an 'abstract idea' or program '*as such*' or 'per se' in each and every software patent case.

While it is important to protect software innovation, it is equally important to limit such protection in scope and length due to the short lifecycle of software production. This will help prevent patent trolls and prompt true software innovation. Allowing the means of implementation but not the function to be patented and providing a 10-years of utility model type or *sui generis* protection are the alternatives to be explored.

The uncertainty over software patenting is not a US problem, but a universal problem confronting every country that has a software industry and a software patenting regime. The unclear definition of 'computer program *as such*', the contradictory rulings from the EPO, and the contradiction between law and practice in China have attested to such uncertainty. Hence, there is a need for a solution beyond national borders. Particularly, because most software patents cover borderless Internet technologies, a solution from international institutions, e.g., a Software Treaty, would be more beneficial to software innovation around the world. Given that it had drafted the Model Provisions and a Treaty for the legal protection of software in the past, WIPO has the expertise and resources needed to tackle this challenging issue. On the other hand, it shall be noted that WIPO faces a mountain of daunting tasks in other areas and the negotiation for the software treaty could be a long and hard journey.