

KnowRight '98
IFIP World Computer Congress 1998
Session B2, Thursday, 3 September 1998, 3:30-5:00 PM

Patent Protection of Computer Software in the United States and Japan

Karl Ruping
<ruping@tuj.ac.jp>

1. Introduction: Patenting Opportunities and Challenges

Does computer software qualify as patentable subject matter? If so, what elements of this new technology can be protected by patent law? With the rise of software applications as a component of domestic output and foreign trade, national patent regimes are struggling with the appropriate extension of protection to software technologies. Despite recent international harmonization of intellectual property rights, wide disparity remains in the area of international software patenting. The US and Japan, for example, offer the broadest protection to software technologies among the industrialized nations while Europe continues to take a more conservative approach.

This paper presents the status of patent protection for computer software technologies in the United States and Japan. I begin with a brief historical look at the software industry and the development of judicial doctrine defining patent software subject matter. A more current review looks to US patent practice and the Japanese analysis of patent subject matter under the government's software patent guidelines. I then focus on the most recent expansion in software patenting: data structures and software programs on a computer readable medium. I conclude that the expansion in software patent scope offers new opportunities for innovators, but poses new challenges to future development in this industry.

2. History of Legal Protection of Software Technology

The development of computer software technologies and the commercial use of resulting products parallels the emergence of a legal regime protecting innovative applications, although with uncomfortable delay. In both the US and Japan this development has been characterized by a migration from trade secret protection to more formal rights under copyright, and recently to the acceptance of innovative software technology as patentable subject matter. Since the early 1980s software patenting has progressively expanded, in the US under liberal court interpretation and in Japan by legislative revision and JPO administrative efforts.

The software patent migration may have advanced too far, with the scope of statutory subject matter in the US and Japan extending to data structures and software applications saved on a computer-readable medium. While these two countries can offer valuable lessons in patent software, broad protection to some technology embodiments should be cautiously considered.

The modern electronic computer emerged in the late 1940s, but did not become commercially attractive until the 1960s. Early software remained the poor cousin of hardware advances. Initially manufacturers maintained incompatible platforms and bundled software with the mainframe computers that comprised this new market. Software bundling began to unravel in the 1970 when IBM openly licensed its proprietary software under US antitrust pressure. Software designers and user groups emerged from the primordial soup of the hardware industry to offer applications that could complement - or compete with - standard software programs. Early attempts to patent these innovative software elements in the US and Japan largely failed. But software technology was still in its infancy both in terms of technological sophistication and economic value.

With the evolution of an independent market the commercial value of software became evident. Bundled software qualified as trade secret or proprietary know how. As program applications became mass-produced and widely available off the shelf,, however, software lost its secret nature and was subject to widespread copying. Trade secret alone was now an insecure form of protection. Following a series of US and Japanese court decisions in the mid 1970's, software providers turned to copyright as the preferred intellectual property right.

The limitations of copyright as the exclusive form of protection soon became apparent. This oldest of intellectual property rights extends to an author's original (in the US) or creative (in Japan) expression. While protecting computer code, copyright does not cover the underlying idea or any expression that has merged with the idea. And in the US and Japan an unauthorized person can still engage in the reverse engineering of software programs by decompiling its code, further limiting copyright as a viable means of protecting software applications.

As computing technology advanced and the value of copyright receded, innovative software developers increasingly sought patent protection, initially with limited success. This effort first began in the US where the software industry was the most advanced, valuable and most vulnerable.

3. US Patent Protection

The US patent holder enjoys a twenty year right to exclude others from working a patented invention. The legislative language establishing patentable subject matter has not change during the emergence of the software industry. Section 101 of the United States Patent Act provides:

Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.¹

Courts have progressively expanded the scope of statutory subject matter to include software as patentable technology. The successful claims usually appear in the form of a process or machine/apparatus, such as a programmed computer. A more recent form, discussed below in section 3.1, is an article of manufacture claim to a data structure embodied on a computer readable medium or software code embodied on a computer readable medium.

3.1 US Supreme Court Case Law

In the early period of the software industry the prospect for patent protection looked bleak. During the 1960s the United States Patent Office (USPTO) generally rejected patent applications covering computer software on the basis of the mental steps doctrine.² The first Supreme Court case to consider a software-related invention under Section 101 was *Gottschalk v. Benson*.³ Benson had sought a process patent to convert binary decimal numbers into computer readable binary numbers through the use of a conversion algorithm. Disagreeing with the examiner's § 101 rejection, the Court of Customs and Patent Appeals (CCPA) interpreted the claims as valid. The Supreme Court reversed, however, and rejected the patent claims as non-statutory subject matter.

The Supreme Court avoided the mental steps doctrine that had earlier preoccupied lower courts and the USPTO. Rather, the *Benson* opinion looked to the underlying mathematical algorithm employed in relationship with other physical or mechanical steps of the claim. The Supreme Court stated:

It is conceded that one may not patent an idea. But in practical effect that would be the result if the formula for converting [binary code decimals] to pure binary numerals were patented in this case. The mathematical formula involved here has no substantial practical application except in connection with a digital computer, which means that if the judgment below is affirmed, the patent would wholly pre-empt the mathematical formula and in practical effect would be a patent on the algorithm itself.⁴

The Supreme Court return to the issue of software subject matter six years later in

¹ Patent Act, 35 U.S.C. § 101. US patent law is based on the Constitutional authority of Congress to establish a system “to promote the progress of science and useful arts, by securing for limited times to ... inventors the exclusive right to their respective ... discoveries.” United States Constitution, Art. I, sec. 8, cl. 8

² The federal courts applied the mental steps doctrine in 1951 to a patent application for a seismic data analysis method used in petroleum prospecting where the only novel elements were the data calculation and interpretation claims. The court stated that a process consisting of entirely mentally executable ideas or operations is not patentable subject matter. *In re Abrams*, 188 F.2d 165 (CCPA, 1951); *see also* Shao Wen Yaun, 188 F.2d 377, 379-80 (CCPA, 1951).

³ *Gottschalk v. Benson*, 409 U.S. 63 (1972).

⁴ *Id.*, pg. 71.

Parker v. Flook.⁵ The *Flook* patent claimed a method for updating values in an alarm system used in the catalytic chemical conversion of hydrocarbons. The only novel element of this invention was the calculation method used to set the alarm limits. Establishing what became the point of novelty approach, the Court opined that the mathematical formula itself could not impart patentability on the claims:

“Respondent’s process is unpatentable under Sec. 101, not because it contains a mathematical algorithm as one component, but because once that algorithm is assumed to be within the prior art, the application, considered as a whole, contains no patentable invention. Even though a phenomenon of nature or mathematical formula may well be known, an innovative application of the principle may be patented. Conversely, the discovery of such a phenomenon cannot support a patent unless there is some other inventive concept in its application.”⁶

Many interpreted the *Benson* and *Flook* cases to mean that software applications which incorporate a mathematical algorithm were not patentable subject matter. This pessimism turned to guarded optimism in 1981 when the Supreme Court held that a man-made microorganism was patentable subject matter.⁷ In the same year the Supreme Court issued the *Diamond v. Diehr*⁸ decision which was to open the gates to software patenting.

Diehr sought a patent for a synthetic rubber curing process which incorporated a programmed computer. The prior art processes calculated the cure time based on a set of initial temperature readings of the mold. Diehr's process used a computer operated program to periodically re-calculate the cure time based on a mathematical computation of changing temperatures during the curing process. Adopting its new expansive interpretation, the *Diehr* Supreme Court stated that statutory subject matter covers “anything under the sun made by man.” But there are important limits to what can be protected:

“The Court has undoubtedly recognized limits to § 101 and every discovery is not embraced within the statutory terms. Excluded from such patent protection are laws of nature, natural phenomena and abstract ideas.”⁹

The *Benson* claim was to an invention that in effect would have precluded any use of a particular mathematical formula and in *Flook* the innovative claim was simply for the computation of data to reach a number, an alarm limit. In *Diehr*, however, the patent was for an improved industrial process incorporating a mathematical computation that resulted in a physical change, the curing of synthetic rubber. This was a claim to the particular process employing the mathematical formula, not the formula itself, and thus patentable subject matter.

⁵ *Parker v. Flook*, 437 U.S. 584 (1978).

⁶ *Id.*, pg. 594

⁷ *Diamond v. Chakrabarty*, 447 U.S. 303 (1981).

⁸ *Diamond v. Diehr*, 450 U.S. 175 (1981).

⁹ *Id.*, pg. 185.

The *Diehr* opinion offered a more inviting interpretation of statutory subject matter, but the limited analysis would soon be eclipsed by continued software development. Following the Supreme Court trilogy, however, judicial doctrine remained based on the existence of a mathematical algorithm hidden in the patent claims.

3.2 Federal Court Interpretation of Patentable Subject Matter.

The subject matter controversy did not end with *Diamond v. Diehr*. The USPTO and courts have attempted to discern the contours of software patent subject matter in an atmosphere of changing technology and vague Supreme Court doctrine rooted in a static technological paradigm. In a series of cases spanning the period of the *Benson-Flook-Diehr* trilogy, the Court of Customs and Patent Appeals (CCPA) attempted to clarify this area of law, but with limited success. The resulting Freeman-Walter-Abele test requires a two-step analysis of the patent claims.¹⁰ The court first determines whether a mathematical algorithm exists, directly or indirectly, in the patent claim. If there is a mathematical algorithm, the court goes on to determine whether the claimed invention, as a whole, is no more than the algorithm itself.

Not all courts employed the Freeman-Walter-Abele test, and some have recognized the difficulty of identifying the existence of a mathematical algorithm in the claims. The successor to the CCPA, the Court of Appeals for the Federal Circuit (CAFC), adopted the prior court's decisions as precedent,¹¹ but in one of its first software patentability decisions the court did not explicitly adopt the Freeman-Walter-Abele test.

*In re Alappat*¹² was a rare *en banc* decision reviewing the USPTO's rejection of a patent application for the manipulation of data to display a smooth waveform on an oscillator. Prior art oscillators would at times display discontinuous waveforms with jagged edges. Alappat's invention employed a mathematical formula that would convert the input data into a set of illumination data of varying intensities to produce a smooth representation of the input data. The court interpreted the claim to be for a machine, a *rasterizer*, capable of converting the input data into corrected illumination data for display based on a mathematical formula.

Alappat's invention included a mathematical formula, but as a whole the claims were to a machine for converting discrete waveform data for display on a screen. This machine – a programmed computer – was distinct from the prior art:

¹⁰ The Freeman-Walter-Abele test emerged from a series of three federal court opinions: *In re Freeman*, 573 F.2d 1237 (C.C.P.A. 1978), decided between *Benson* and *Flook*; *In re Walter*, 618 F.2d 758 (C.C.P.A. 1980), decided after *Flook*; and *In re Abele*, 684 F.2d 902 (C.C.P.A. 1982), decided after *Diehr*.

¹¹ *South Corp. v. United States*, 690 F.2d 1368 (Fed. Cir. 1982).

¹² *In re Alappat*, 33 F.3d 1526 (Fed. Cir. 1994).

programming creates a new machine, because a general purpose computer in effect becomes a special purpose computer once it is programmed to perform particular functions pursuant to instructions from program software.¹³

The *Alappat* court did not declare the Freeman-Walter-Abele test invalid. Citing the difficulty in identifying a mathematical algorithm,¹⁴ however, the CAFC returned to the language of the Supreme Court in *Diehr*: Patent protection does not extend to laws of nature, natural phenomenon and abstract ideas. The proper inquiry in dealing with mathematical subject matter is to see whether the claimed invention as a whole is a disembodied mathematical concept beyond the scope of statutory subject matter.

3.3 US Computer Guidelines

In 1996 the USPTO issued the Examination Guidelines for Computer Related Inventions.¹⁵ These are intended to be a summarization of prior case law such to provide examiners with a framework from which to consider the patentability of software patent applications. The Computer Guidelines do not constitute substantive rules binding on applicants, the Patent Office or the courts, and an examiner's failure to follow these provisions is neither grounds for an appeal nor a subject for petition.

The Computer Guidelines call for the examiner to classify the invention as a process, machine, or article of manufacture. A machine claim is directed to a computer or other programmable apparatus, such as Alappat's rasterizer. A process claim is directed to a series of specific operations or machine implemented steps, such as Diehr's computer-implemented curing process. A new category, discussed in detail below, is an article of manufacture claim to a computer-readable medium incorporating a software program or a computer-readable medium encoded with a data structure.

The Software Guidelines were not intended to have judicial weight, but some courts have viewed the guidelines as persuasive authority.¹⁶ In the general terms of apparatus and process claims, the Computer Guidelines are in conformance with prior case law. This is an exercise in interpretation, however, and the contours of statutory subject matter remains unclear. The touchstone of patent subject matter remains §101, as interpreted by the Supreme Court trilogy and subsequent federal court decisions. While the Computer Guidelines have been a source of clarification - and even comfort - for those drafting software claims, the general themes expressed in case law remain controlling and are critical reference points as software technology continues to push at

¹³ Id., pg. 1545. A recent return to the Freeman-Walter-Abele test is *State Street Bank v. Signature*, 927 F. Supp. 502 (1996), under appeal as of this writing.

¹⁴ In a footnote to the *Alappat* decision the CAFC noted: "The Supreme Court also has not been clear as to exactly what kind of mathematical subject matter may not be patented. The Supreme Court has used, among others, the terms 'mathematical algorithm,' 'mathematical formula,' and 'mathematical equation' to describe types of mathematical subject matter not entitled to patent protection standing alone." In re Alappat, 33 F.3d 1526, 1543, note 19 (Fed. Cir. 1994).

¹⁵ Examination Guidelines for Computer-Related Inventions, 61 Fed. Reg. 7478 (1996).

¹⁶ In re Trovato, 60 F.3d 807 (Fed. Cir., 1995).

the outer edges of patent subject matter. In Japan, however, the JPO's Computer Guidelines play a much more important role in the apparent legal hierarchy.

4. Japan Patent Protection

The US software subject matter debate is based on a rich history of case law with the USPTO reacting to, rather than leading, judicial interpretation of constitutional and legislative text. In Japan it has been the government bureaucracy that has determined software patentability, in particular the Japan Patent Office (JPO). This is consistent with the civil law traditions and the Germanic patent foundation which has been dominant since Japan's earlier Patent Act of 1899.

Under the current Patent Act of 1959¹⁷, as revised, the holder of a Japanese patent enjoys the exclusive right to commercially work a protected invention for a period of twenty years from the date of filing the patent application. A patent is granted only to subject matter which falls within the statutory definition of invention (*hatsumei*). Japanese patent law limits patentable inventions to a highly advanced creation of a technical idea utilizing a law of nature (*shizen hoosoku*).¹⁸ While an ambiguous definition, the JPO has progressively expanded the interpretation of patentable invention in the area of software consistent with US judicial developments, and apparently modeled its own patenting provisions after the US Software Guidelines.

4.1 Early Legal Analysis

Even before the courts had an opportunity to comment, the JPO issued its first statement on computer software patents in 1975 with the Examination Standards for Computer Program Related Inventions.¹⁹ Similar to the pessimistic impression left by the Benson and Flook cases, early JPO policy appeared to reject software as patent subject matter, and chilled attempts to claim software-related inventions. Software innovations instead depended on copyright protection.

In Japan two ministries have long competed for administrative control over the computer software industry. The Ministry of Education and its Cultural Affairs Agency (CAA) is responsible for the copyright system while the Ministry of International Trade and Industry ("MITI") administers patent, utility model, and trademark registration through the JPO. In the early 1970's, as a domestic software market struggled to emerge, the CAA and MITI initiated competing studies concerning the appropriate protection of computer software. The 1972 MITI proposal was for a *sui generis*

¹⁷ Japanese Patent Act, Law No. 121 of 1959.

¹⁸ Japanese Patent Act, Sec. 2(1). While protection applies to all areas of technology except when the invention is contrary to public order, good morals or public health.

¹⁹ Examination Standards for Computer Program Related Inventions, Part I, issued December 1975. The 1975 Examination Standards were followed in 1982 with the Examination Guidelines on Inventions Related to Microcomputer-Applied Technology. The Microcomputer Guidelines did not replace the 1975 Computer Program Guidelines, which remained in force, but neither did the Microcomputer Guidelines clarify the issue of software patenting.

registration-based law under the administration of that ministry. The subsequent CAA report supported the extension of copyright to software programs. Failing to reach a consensus between these two ministries, the government undertook no legislative activity to clarify the legal protection of software technology.

While the bureaucracy remained deadlocked, Japanese computer games emerged as an internationally competitive product and copyright infringement cases began to appear before the courts. The usually restrained Japanese judiciary first recognized the copyrightability of computer programs in the 1982 *Taito Case*.²⁰ In response to judicial developments, both ministries initiated new studies concerning the legal protection of computer software. In a 1984 report the CAA maintained that copyright was the emerging international form of protection and that the Copyright Act should explicitly recognize domestic cases holding that software code qualifies as copyrightable material. Soon thereafter a MITI report proposed a Program Right Law which looked to the Patent Act as a model. The ministries reached an informal agreement in early 1985 and the Diet adopted a revised Copyright Act recognizing software copyrightability, effective as of January 1, 1986.²¹ But software patenting remained an issue within the JPO rather than other government agencies, including the Ministry of Justice and the courts.

4.2 JPO's 1993 Computer Guidelines.

The scope of Japanese copyright does not extend beyond the literal software code. Copyright could provide only limited protection of software technology and interest turned to patent protections. Criticism of the early Patent Guidelines emerged as computer technologies developed and the domestic software market expanded rapidly in the early 1990s. The JPO responded with the 1993 Examination Standards for Computer Software-Related Inventions. These new provisions recognized that even if an element of the invention does not qualify as statutory subject matter alone, the invention may still be patentable subject matter if the invention as a whole is a technical idea that utilizes a law of nature.

The 1993 Software Guidelines did not go as far as the *Diehr* Court in recognizing a broad scope of subject matter under Japanese patent law. Practitioners and scholars remained dissatisfied. While it was now clear that the existence of a mathematical algorithm as an element of the claim will not render the invention as non-patentable subject matter, the 1993 Software Guidelines did not explain when a software-related invention as a whole will qualify as incorporating a technical idea and utilizing a law of nature. No cases emerged to clarify the relationship between software

²⁰ Taito developed a video game entitled Space Invader Part II. The Tokyo District Court held that the exclusive right of reproduction was infringed when the defendant, Makoto Denshi Kegi, copied the entire source code from the read-only-memory or a Taito video game machine. Judgment of Dec. 6, 1982, Tokyo Chisai (Dist. Ct., Tokyo), 1060 Hanji 18 (1983). The Yokohama and Osaka District Courts reached similar findings in 1983 and 1984, respectively. Judgment of Jan. 26, 1984, Osaka Chisai (Dist. Ct., Osaka), 1106 Hanji 134 (1984) (Konami Kegi v. Daiwa); Judgment of Mar. 30, 1983, Yokohama Chisai (Dist. Ct., Yokohama), 1081 Hanji 125 (1983) (Taite v. Makoto Denshikegi).

²¹ Partial Amendment of the Copyright Act, revised art. 10 (Law No. 62, 1985).

technology and the technical idea/law of nature requirement. While more receptive to software patenting than the previous guidelines, the new provisions appeared to be nothing more than an incorporation of US case law absent a full consideration of the developing technology and domestic patent doctrine.

4.3 JPO's 1997 Revised Computer Guidelines

Following extensive research efforts the JPO issued a revised statement on computer software patenting in 1997. Again, the revised JPO Software Guidelines followed developments in US case law, but also incorporated the new US Software Guidelines. There are three specific types of software claims: a process claim directed to a series of operations; a product claim directed to one or more functions; and a memory claim which includes a computer-readable medium used in the functioning of hardware or a computer-readable medium encoded with a data structure.

The status of the JPO Software Guidelines relative to their US counterpart is difficult to establish. However, they appear to have relatively more judicial weight for several reasons. First, there is no case law on point. Courts have not had an opportunity to develop a judicial doctrine of software patenting from which to establish a line of reasoning different from the well established guidelines. Second, under the civil law nature of the Japanese legal tradition, courts are likely to defer to the expertise of the JPO, developed over 23 years of software policy.

Finally, the JPO enjoys the legislative authority to initially determine statutory subject matter, with appeal review by the Tokyo High Court and eventually the Supreme Court of Japan. Usually it would be a patent rejection that takes this review path, with positive grants of patent protection to questionable subject matter challenged only at the time of enforcement. Even then the judicial challenge is in the guise of scope of patent right rather than statutory subject matter review of the JPO examination.²² Thus, one can predict that Japanese courts will give significant deference to the Software Guidelines if a decision was ever to be handed down.

5. Recent Expansion: Article of Manufacture Claims

The most recent expansion in US and Japanese software patenting has been in the area of article of manufacture claims. In the past software-related inventions were usually claimed either as a process/method or a machine/apparatus. Software developers remained frustrated by the limited scope of enforcement powers available under such claims. For a patented process to be infringed the unauthorized party must have worked the process. Producing or distributing a software program capable of performing the process or a step in the process was not sufficient to establish infringement. Similarly, a machine claim required not only the software code but also the claimed hardware element to establish infringement, again leaving the sale and distribution of software

²² For an interesting comparative analysis of Japanese patent scope see Toshiko Takenaka, *Interpreting patent Claims: The United States, Germany and Japan* (IIC Studies, 1995)

alone beyond the patent holder's control. When direct infringement fails an alternative is to allege contributory or induced infringement. Indirect infringement theories, however, require a heightened level of evidentiary proof that includes knowledge of the intended infringing use.

The introduction of article of manufacture claims to the family of software patenting transfers non-infringing - or indirectly infringing - activities to the category of direct infringement. The sale or distribution of software programs on a floppy disk, CD-ROM or other computer readable medium will infringe the appropriately drafted article of manufacture claim.

The excitement and controversy over article of manufacture software patents began in the US with an application claiming innovative data structure elements. The *In re Lowry* claims were drawn to a data structure which managed the storage properties of information saved in memory using attributive data objects (ADO).²³ While there was no visible structure *per se*, the invention was to electrical or magnetic structural elements which imparted a physical organization on the collection of bits stored in memory. The CAFC established in *Lowry* that data structures which 1) require computer processing and 2) impart structural organization on data in memory qualify as statutory subject matter.

With data structures under the umbrella of patent protection, IBM had sought to test the increasingly warm waters of software patenting by claiming a computer program in the form of an article of manufacture. The *In re Beauregard* patent was to an article composed of a computer useable medium embodying a program code which implemented a method for filling complex shapes in a graphic display device, a computer screen. The claim was rejected and IBM appealed this rejection to the CAFC. Before the court could consider the claims, however, the USPTO reconsidered its initial rejection and held the claims to be patentable.

To understand *Beauregard* we must return to *Lowry*. The USPTO's initial rejection of both the graphic software program and the ADO data structure was based on the printed matter doctrine. A narrow line of cases established that information on a substrate is not patentable subject matter unless there is a novel, non-obvious functional relationship between the information and substrate.²⁴ The USPTO found the data structure (information) on a computer memory (substrate) was equivalent to printed matter. The CAFC reversed, however, holding that the printed matter doctrine was limited to arrangements of printed lines of characters useful and intelligible only to the human mind. The *Lowry* claim required processing by a machine rather than a person and, further, demonstrated electrical or magnetic structural elements imparting a physical organization to the data stored in memory.

²³ *In re Lowry*, 32 F.3d 1579 (Fed. Cir. 1994). Several days prior to *Lowry* a different panel of the CAFC rejected a data structure alone, but upheld a machine with a data structure memory as statutory subject matter. *In re Warmerdam*, 33 F.3d 1354 (Fed. Cir. 1994).

²⁴ *In re Gulack*, 703 F.2d 1381 (Fed. Cir. 1983); *In re Russell*, 48 F.2d 668 (CCPA 1931).

With the defeat of the printed matter doctrine under *Lowry*, IBM's challenge to the Board's rejection of the *Beauregard* claim looked promising, but by no means secure. First, the software code saved on the computer medium is a set of written instructions in a series of 0s and 1s to fill in a polygon. The court did not determine if the software code contained structural elements or imposed a physical organization on the memory element.

With these questions still lingering and before the CAFC could consider the *Beauregard* appeal the commissioner of the USPTO withdrew the Board's rejection in light of *Lowry*. The CAFC accordingly dismissed the appeal, depriving the software community of clear guidance on software article of manufacture claims. Despite the absence of a judicial statement on point, the USPTO drafted its Software Guidelines to include *Beauregard* software claims in addition to *Lowry* data structure claims. Article of manufacture claims extend to functional descriptive material which imparts functionality on memory or hardware resources when encoded on a computer readable medium. Non-functional descriptive material such as a music recording or a mere arrangement of data stored on a computer memory does not qualify for patent protection.

Japanese courts have not faced a data structure or software article of manufacture challenge and still has yet to issue an opinion on software subject matter. Regardless, the JPO has been active in expanding the scope of statutory subject matter consistent with the USPTO Software Guidelines even when these extend beyond existing US case law. With the 1997 revisions the JPO Software Guidelines now recognize data structures embodied on a computer readable medium and software code embodied on a computer readable medium as patentable inventions. The claims must still sufficiently described a novel, non-obvious advance in the particular field of technology relative to the existing technology, the prior art.

6. Software Patenting Issues

Software patenting provides innovators with a chance to protect their inventions, but also raises new concerns for the future of the software industry. The issue of patentable subject matter has been partly clarified by case law in the US and by Patent Office guidelines in Japan. With these clarifications and the expanding scope of protection, the volume of software patent applications have increased. But the US Supreme Court has not taken the opportunity to check recent judicial and USPTO expansion in software subject matter since its 1981 *Deihr* decision. An in Japan there has been no judicial statement on the expansive JPO software policy.

One concern is uncertainty. Article of manufacture claims drawn to a computer readable medium provide patent protection to functional descriptive material under both US and Japanese guidelines. The divide between functional and non-functional descriptive material is blurred at best, however, and may not be able to withstand continued technical advances. A further uncertainty is in the definition of a computer

readable medium. Floppy discs and CD-ROMs do not pose a conceptual problem, but a fiber optic line or satellite transmission may form the basis for a propagated signal claim – a computer data signal embodied in a carrier wave. And as the internet becomes the software distribution system of choice, non-physical or ephemeral transfer devices are a new focus for software patent claims.

A further concern is the validity of issued patents and the scope of their claims. Software technology was initially protected by trade secret law, which still continues as one possible form of protection. Copyright remains the primary intellectual property right protecting software code, and requires no registration or filing procedures. With this history of trade secret and copyright protection, access to non-patent prior art remains a difficulty for new software developers and patent examiners. Both the USPTO and JPO are issuing patents of questionable validity with limited prior art citations. This introduces uncertainty into the patent field, and allows for potential abuse in terms of enforcing suspect patents against alleged infringers who would rather settle than face a long and expensive court battle.

7. Conclusion.

As software technology becomes increasingly sophisticated and the value of more expansive applications becomes apparent, the number of software patents will continue to increase. And as this new form of technology develops, so will the legal theories of patent subject matter and patent scope.

Both the US and Japan have adopted similar tests of software patentability, but based on very different decision mechanisms and legal traditions. In the common law US a long history of judicial decisions has dynamically interpreted the constitutional and legislative grant establishing patent subject matter. In civil law Japan it is the bureaucracy – in particular the JPO – that has drafted guidelines on patent subject matter. While the Japanese courts have yet to explicitly establish the parameters of software patents, the internal rules for patent examiners is likely to remain the foundation of applicant strategy and court analysis.

It is not clear if this substantive harmony between the two patent systems will continue as software technology develops further in light of the disharmony of legal traditions and procedures. A further question is whether this model will form the basis for other national patent systems, namely Europe. Continued efforts at joint discussion between the two patent offices, and among participants of the international stage, will be critical in ensuring not only a common approach to this issue, but also the preferred approach.