

Robert A. Gehring und Martin Kretschmer

Software-Patente in Vergangenheit, Gegenwart und Zukunft

schwerpunkt

Patentschutz für Software ist, historisch betrachtet, das Ergebnis einer recht jungen, kaum mehr als 40 Jahre alten Entwicklung. Ihr voraus ging die Emanzipation der Software von den schaltungstechnischen Rechenungetümen der 40er Jahre, deren Programme sich weniger in Kilobyte denn in Kilogramm Schaltungsaufwand beschreiben ließen. Der Wirtschaftsaufschwung der 60er Jahre brachte dann eine stetig steigende Nachfrage nach Rechenleistung mit sich: die zivile Computerindustrie begann aufzublühen, sich zur profitablen Industrie zu entwickeln.

Erfolgt die frühen Software-Innovationen noch ohne jeden Rechtsschutz, also unter scharfen Wettbewerbsbedingungen, wuchs bald der Wunsch nach staatlichem Schutz von Investitionen – getreu dem bekannten Muster anderer Industriezweige. Wurde zunächst der starke Ideenschutz des Patentrechts abgelehnt zugunsten einer Verankerung des Codes im Urheberrecht, nähern wir uns heute dem faktischen Doppelschutz. Im vorliegenden Aufsatz¹ beleuchten die Autoren einige Meilensteine auf dem Weg zur Patentierbarkeit von Software.

Die 50er Jahre – Software emanzipiert sich

Zur Erinnerung: Die 40er Jahre hatten einen Quantensprung in der Entwicklung der Rechenmaschinen mit sich gebracht. Babba-ges alter Traum² von der programmierbaren Universalrechenmaschine stand kurz vor der Erfüllung. Nach dem Ende des Zweiten Weltkriegs wurde die Entwicklung international vorangetrieben. Der zunächst erleichterte Informationsaustausch zwischen Universitäten verschiedener Länder, und innerhalb einzelner Länder, sowie die nun nicht mehr vom Krieg absorbierten Finanzmittel ermöglichten es neue Konzepte umzusetzen.

Neben vielen anderen spielten Eckert, Mauchly und von Neumann eine Schlüsselrolle bei der Verbreitung der Idee des modernen Universalcomputers. Während Eckert und Mauchly, beraten von Patentanwälten (Macrae 1994, S. 248), vorrangig versuchten mit ihrer Eckert-Mauchly Computer Corporation Geld zu verdienen,³ blieb John von Neumann ein Mann der Wissenschaft. Er sprach sich vehement gegen restriktive Patente auf Computertechnologie aus, die er als Innovationshindernis betrachtete (Macrae 1994, S. 257). Seine während der gemeinsamen Forschungen mit Eckert, Mauchly und ihrem Team entwickelten Vorstellungen zur Weiterentwicklung des Computers brachte er 1945 zu Papier,⁴ dessen nachfolgende Publikation und restriktionsfreie Verbreitung maßgeblich zur Wirkung der von Neumann'-Architektur des modernen Universalcomputers beigetragen haben dürfte.⁵ Weitere seiner Artikel aus der Zeit in Princeton regten

„[v]erschiedene Institutionen [dazu an] – mit Johnnys Zustimmung – „Raubkopien“ des vorgeschlagenen IAS-Computers zu machen, während sich dieser noch im Bau befand, und sie arbeiteten manchmal schneller als das kleine Team am IAS. Überall führte jede Lokalgröße bei

jedem Projekt eigene Modifikationen ein. Das erwies sich als Königsweg zum Beginn einer neuen Technologie, und es ist schade, daß er seitdem nicht öfter [...] beschritten worden ist.“ (Macrae 1994, S. 267)

In den 50er Jahren etablierte sich eine Anzahl von Unternehmen als Lieferanten der neuen Rechentechnik. Die entsprechend der ‚von Neumann'-Architektur konstruierten Computer zeichneten sich durch einen variabel zu füllenden Speicher aus, der Programme und Daten nach Belieben aufnehmen konnte. Dank der Fortschritte in der Hardwaretechnologie (Elektronenstrahlröhren) vereinfachte sich die Eingabe der Programme und Daten erheblich. Anstelle von Relaiszuständen (also Hardware) wurden jetzt Elektronen in Röhren zum Speichern genutzt – Software hatte ihr Embryonalstadium erreicht.

Das Programmieren ähnelte immer weniger dem vollständigen Umbau des Gerätes, was einen deutlichen Zeitvorteil bei Vorbereitungen von Berechnungen mit sich brachte. Höhere Rechenleistung und vereinfachte Programmierung wiederum ließen die Nachfrage aus Wirtschaft und Verwaltung ansteigen: a virtuous cycle.

Die 60er Jahre – Begründung der Software-Industrie

Was in den 50er Jahren begonnen hatte, erlebte in den 60er Jahren ein stürmisches Wachstum. Computer wurden Universalmaschinen nicht nur in der Architektur, sondern auch in der Verbreitung, das eine die Folge des anderen. Die Programmierung bestand nunmehr aus der Eingabe von Code, in der Regel gespeichert auf Karten (später Loch- und Magnetbändern), und Software war auf dem Weg textuellen Charakter anzunehmen. Die neuen Computer ließen sich zu Berechnungen jeder Art einsetzen und drangen daher überall dort schnell vor, wo ein hoher Bedarf an korrekten Berechnungen vorhanden war: Militär, Behörden und Wirtschaft. Während Militär und Behörden in der Regel Spezialanwendungen im Einsatz hatten, ähnelten sich die Prozesse in der Wirtschaft stark. Die Bedürfnisse einer seit Jahrhunderten standardisierten Buchhaltung erleichterte den Computerherstellern die Arbeit ungemein: Sie lieferten ein- und dieselbe Technologie an unterschiedlichste Kunden aus. Damit einher ging natürlich auch eine Standardisierung der Herstellungsprozesse, die wiederum mit Kostensenkungen verbunden

war. Computer wurden daher relativ schnell immer billiger, was die Nachfrage befeuerte und die Verbreitung vorantrieb.

Anfang der 60er Jahre erwarb man Software nur als Zugabe zur Hardware.⁶ Das Geschäft bestand im Verkauf von Hardware im Bündel mit einigen kostenlosen Standardanwendungen. Niemand kam auf die Idee, Software separat anzubieten, dafür schien der Markt einerseits zu klein und andererseits zu zersplittert. Stattdessen bildeten sich Dienstleistungsunternehmen heraus, die Programmierung als kostenpflichtigen Service anboten.⁷ Hinzu kam natürlich die Softwareentwicklung für den Eigenbedarf in den Unternehmen selbst.

Die kleinen Dienstleister bildeten den Keim der sich entwickelnden Softwareindustrie:

„Most such companies were small, but a handful were large enough to go public, employing hundreds of programmers. These firms increasingly found opportunities to package the software they had already written and deliver it to multiple customers, a situation that promised potentially high profits given the low cost to reproduce already developed software. The term software packages appeared in the late 1960s and implied that the customer deliverables included documentation and some level of service, such as installation, as well as the program code. Many early products were utility programs with greater functionality or efficiency than the comparable free software from the hardware vendors. Other early products were software applications like payroll or banking where external factors such as government regulations imposed a uniformity on the way that customers defined their specifications. In January 1967, International Computer Programs (ICP) in Indianapolis, Indiana, began publishing a quarterly catalogue of computer programs available for sale, and the software product industry began to take shape.“
(Johnson 2002, S.14)

Der entstehende Wettbewerb zwischen den Unternehmen wurde in den späten 60er Jahren angeheizt durch den Druck der Wettbewerbshüter auf die großen Anbieter, Software und Hardware entbündelt zu vertreiben (Branscomb 1994, S. 141). Es wurden daher Begehlichkeiten nach staatlichem Schutz für Software geweckt, um sich gegen unerwünschte Wettbewerber einen Marktvorteil zu verschaffen. Patentschutz als das stärkste Schutzinstrument erschien etlichen Unternehmen als angemessen und so begannen sie Patente auf Verfahren anzumelden, die üblicherweise in Form von Software implementiert wurden. Vorreiter waren auch hier U.S.-Unternehmen, wie der Fall Gottschalk v. Benson dokumentiert.

Die Bell Labs meldeten 1963 ein Verfahren zur Konvertierung von binär codierten Dezimalzahlen (BCD) in reine Binärzahlen zum Patent an. Das US-Patentamt mochte allerdings nicht einsehen, dass es sich um eine patentwürdige Erfindung handle, vielmehr erkannte man darin mentale und mathematische Prozesse, die nicht patentierbar seien. Mit dieser Reaktion unzufrieden klagte Bell sich durch die Instanzen bis zum Supreme Court. Im Jahr 1972 fiel schließlich das Urteil (Gottschalk, Commissioner of Patents v. Benson; 409 U.S. 63, 1972). Das Gericht befand in seinem vielkritisierten Urteil, dass der Patentanspruch

einem Algorithmus gelte und diese seien nicht patentierbar. Nichtsdestotrotz wurden Patentansprüche auf Software nicht grundsätzlich ausgeschlossen.

In seinem Urteil berief sich das Gericht ausdrücklich auf die Empfehlungen einer vom Präsidenten eingesetzten Kommission zum Patentsystem, die sich 1966, unter Verweis auf die erfolgreiche Entwicklung der Softwareindustrie auch ohne Patentschutz, gegen solchen für Software ausgesprochen hatte. Stattdessen sollte der bereits etablierte Copyright-Schutz für Software das Schutzinstrument der Wahl sein. Seit 1961 hatte das U.S. Copyright Office Kopien von Programmtexten zur Hinterlegung angenommen und somit deren Copyright-Schutzfähigkeit grundsätzlich bestätigt. Im Kern galt »Software is Unpatentable«, wie es in einem Leitartikel der New York Times von 1968 hieß (Jones 1968).

Die 70er und 80er Jahre – Persönliche Computer und frühe Vernetzung

Auf der technischen Seite brachten die 70er Jahre den Mikroprozessor (Intel 1971) und, wenige Jahre später, nicht nur den Microcomputer, sondern auch die ersten persönlichen Computer (MITS Altair, 1975; Apple II, 1977; IBM PC, 1981), die nicht mehr nur für Geschäftsanwender, sondern ausdrücklich für Heimanwender im Markt positioniert wurden. Der Erfolg insbesondere des IBM PC brachte es mit sich, dass in immer mehr Büros, auf immer mehr Schreibtischen Computer mit einer Rechenleistung standen, die wenige Jahre zuvor nur für wenige Unternehmen erschwinglich waren. Datenerfassung und Datenaustausch wurden zum nicht mehr wegzudenkenden Bestandteil der Büroarbeit, oft genug zum strategischen Faktor des Unternehmenserfolges. Aber auch bei zivilen Behörden und beim Militär wuchs die Abhängigkeit von der Rechentechnik, vom funktionierenden Datenaustausch. Zu dessen Erleichterung und Sicherstellung lag es nahe die Vernetzung der verteilten Rechenkapazitäten voranzutreiben. Die Defense Advanced Research Projects Agency (DARPA) gab Mitte der 70er die Entwicklung eines ausfallsicheren Netzwerkes in Auftrag. Wir wissen, wohin das führte: Die TCP/IP-Protokollfamilie bildet heute das Rückgrat des Internets.

Um der wachsenden Marktbedeutung von Software gerecht zu werden, wurden 1976 der U.S. Copyright Act dahingehend geändert, dass Software als literarischem Werk ausdrücklich Schutz zugestanden wurde: »[T]he expression adopted by the programmer is the copyrightable element in a computer program, and [...] actual processes or methods embodied in the program are not within the scope of the copyright law,« wie es in einem Bericht des Repräsentantenhauses heißt.⁸ Wieweit der Schutz genau reichen sollte, blieb offen und die Frage wurde an eine Commission on New Technology Uses of copyrighted works (CONTU, 1975-1978) weitergereicht. Dort wurde schließlich bestimmt, dass Ablaufdiagramme, Quellcodes und Objektcodes Copyright-Schutz genießen sollten. Der ebenfalls geprüfte Vorschlag, ein gesondertes Schutzrecht (sog. *sui generis* Recht) für Software zu schaffen statt das alte Copyright der neuen Technologie überzustülpen, wurde hingegen abgelehnt.

Ein Spionagefall in den frühen 80er Jahren zwischen IBM (USA) und Hitachi (Japan) erwies sich dann als Schlüssel für die In-

ternationalisierung des Softwareschutzes. Hitachi und Fujitsu erklärten sich in der Folge des unrechtmäßigen Erwerbs von IBM-Unterlagen zu Lizenzzahlungen an IBM bereit. Grundlage dafür war das Copyright, das IBM an den Unterlagen, darunter Softwarepläne, geltend machte. Die japanische Administration wurde aufgeschreckt und beeilte sich ein eigenes Schutzrecht für Software auf den Weg zu bringen, das Folgeentwicklungen ohne Zustimmung des originalen Softwareentwicklers ermöglichen sollte. Statt dem U.S.-Ansatz zu folgen, konzipierte man ein „sui generis“-Schutzrecht mit Softwareregistrierung, maximal 15 Jahren Schutzdauer und Zwangslizenzen.

Sowohl die U.S.-Regierung als auch die EU-Kommission empfanden den japanischen Sonderweg als Affront und intervenierten. Die japanische Regierung gab nach und verabschiedete Schutzbestimmungen nach U.S.-Vorbild. Dem schloss sich die EU-Kommission mit der Verabschiedung der Softwaredirektive (91/250/EC) an. Die Globalisierung des U.S.-Softwareschutzes fand ihren Abschluss in diversen internationalen Verträgen: NAFTA (1993), TRIPS (1994) und WCT (1996).

Den Kopierschutz, den das Urheberrecht den Softwareunternehmen gewährte, genügte vielen aber nicht. Sie fuhren, oft mit Erfolg, in den 70er Jahren fort Patente auf Verfahren anzumelden, die regelmäßig in Software implementiert wurden. Die Befürworter des Patentschutzes beriefen sich auf die Vorzüge der Offenlegung von innovativen Funktionsprinzipien durch Patentanmeldung, wohingegen binär verteilte Programme aus ihren inneren Funktionen ein Geheimnis machen würden.⁹

Die Gerichte folgten in den späten 70ern den skeptischen Patentämtern und zeigten sich wenig aufgeschlossen in Software gegossenen mathematischen Verfahrensregeln Patentschutz zu gewähren.¹⁰ Wie groß der Andrang bei den Patentämtern trotzdem gewesen sein muss, lässt sich daran ermesen, dass trotz aller Widerstände das U.S.-Patentamt jährlich hunderte Software-Patente erteilte, wie es die Statistiken ausweisen.

Eine vorläufige Klarstellung brachte der vorerst letzte Softwarepatente-Prozess vor dem U.S. Supreme Court, der Fall Diamond

v. Diehr (450 U.S. 175, 1981). In einer knappen 5:4 Entscheidung erging der Bescheid, dass Formeln oder Algorithmen – in diesem Fall die Arrhenius-Gleichung – als Bestandteile eines technischen Prozesses kein Hindernis für die Patenterteilung seien:

„Their process admittedly employs a well known mathematical equation, but they do not seek to pre-empt the use of that equation. Rather, they only seek to foreclose from others the use of that equation in conjunction with all the other steps in the claimed process.“ (S. 185f)

Unternehmen vernahmten die Botschaft und formulierten in Zukunft ihre Anmeldungen für Software-Patente als Prozesse. Wird nicht ein Universalcomputer durch Software in neuer, erfinderischer Weise patentwürdig rekonfiguriert?

In Europa folgte man den in den USA vorgegebenen Linien mit einiger Verspätung. Das Europäische Patentamt (EPO bzw. EPA) entschied 1987 im Falle Vicom (EPO Board of Appeal, T208/84), dass ein Verfahren zur Bildbearbeitung mittels Computer patentfähig sei. Ursprünglich hatte Vicom ein Verfahren zur zweidimensionalen Datentransformation ohne Angabe eines spezifischen Zweckes zum Patent angemeldet. Das war vom Patentamt zurückgewiesen worden, wogegen Vicom Einspruch einlegte. Im Berufungsverfahren wurde Vicom dann gestattet den Anspruch auf Bildbearbeitung einzugrenzen, um ihn patentfähig zu machen (da die Daten ein physikalisches Objekt, eben das Bild, repräsentieren würden).

Nach der Vicom-Entscheidung hat das EPA seine Erteilungspraxis dahingehend liberalisiert, dass »all inventions that might reasonably be considered as within the realms of computer science, for example procedures at the operating system level to improve machine operation, or generic algorithms, techniques and functionality at the application level« (Davies 2003) patentfähig wurden.

Die 90er Jahre – Das Internetzeitalter beginnt

Das seit Anfang der 90er Jahre schnell wachsende Internet brachte massive Umwälzungen der Kommunikationskultur mit sich. Weltweite Datenübertragungen in Sekundenschnelle wurden von der Ausnahme zur Regel, E-Mail löste den Briefverkehr ab. Und spätestens mit dem Auftritt des World Wide Web begann die informationstechnische Eroberung der Haushalte in den industrialisierten Ländern. Nichts davon wäre ohne Software möglich gewesen und konsequenterweise explodierten die Umsätze der großen Softwarefirmen. Das ermutigte immer neue Wettbewerber zum Markteintritt und die Sitten im Umgang miteinander wurden rauher. Jeder versuchte seinen Teil vom Umsatzkuchen so gut es ging, rechtlich abzusichern und Softwarepatente spielten eine große Rolle dabei.

Die wachsende Bedeutung von Software und anderen Informationsgütern in der Zukunft erkennend hatten die in der WTO organisierten Staaten 1994 das Abkommen zu handelsbezogenen Aspekten des Geistigen Eigentums (TRIPS) unterzeichnet, welches in Artikel 27 (1) fordert, dass

Über die Autoren

Robert A. Gehring

Studium der Elektrotechnik, Informatik und Philosophie in Berlin und Ilmenau. Während des Studiums Tätigkeit als freier Consultant, Autor und Dozent; zur Zeit wissenschaftlicher Mitarbeiter im Fachgebiet Informatik und Gesellschaft der TU Berlin am Lehrstuhl von Prof. Lutterbeck; Forschungsschwerpunkte: Freie Software/Open Source, geistiges Eigentum und IT-Sicherheit.

Kontakt: rag@cs.tu-berlin.de

Martin Kretschmer

PhD, zur Zeit Vorstandsmitglied im „Centre for Intellectual Property Policy & Management“, Lehrstuhl für „Information Jurisprudence“ an der „School of Finance & Law“, UK. Verschiedene Gastprofessuren, u. a. 2003 Gastdozent am Wissenschaftszentrum Berlin.

Kontakt: mkretsch@bournemouth.ac.uk

»patents shall be available for any inventions, whether products or processes, in all fields of technology, provided that they are new, involve an inventive step and are capable of industrial application.«

Patentjuristen in aller Welt, nicht zuletzt solche in Diensten großer Softwarehersteller, legen diesen Passus so aus, dass davon auch Software erfasst sei.

Auch das U.S.-Patentamt passte seine Richtlinien dem Geist der neuen Zeit an und gibt seit 1996 Hilfestellung, wie Software zu patentieren sei: Die Patentansprüche sollten sich auf eine Maschine statt auf ein Verfahren oder eine Methode richten. Dass es sich dabei auch um einen Universalcomputer handeln könne, sei dann kein Hindernis mehr bei der Patenterteilung.

Vor einem U.S.-Berufungsgericht wurde 1998 das Urteil in der Sache *State Street vs. Signature Financial Group* (149 F.3d 1368) gefällt. Streitgegenstand war ein Patent auf eine Berechnungsmethode, das Signature Financial erteilt worden war (U.S. Pat. 5,193,056). Üblicherweise sind mathematische Algorithmen von der Patentfähigkeit ausgeschlossen, entweder durch den Wortlaut des Gesetzes und/oder die Prüfungsrichtlinien der Patentämter. Dass es zwischen einer Berechnungsmethode und einem mathematischen Algorithmus einen so großen Unterschied geben soll, ist dem Normalsterblichen jedenfalls nicht ohne weiteres einsichtig. Das U.S.-Patentamt hatte, nach kreativer Interpretation der Patentierungsvoraussetzungen, Signature Financial trotzdem ein Patent auf eine Berechnungsmethode erteilt, wogegen der unmittelbare Konkurrent *State Street Group* vor Gericht zog.

Das Berufungsgericht bestätigte das Patent für Signature Financial mit der Begründung, dass es sich nicht um »merely abstract ideas constituting disembodied concepts or truths that are not 'useful'« (S. 1373) handeln würde und schließlich seien nur solche von der Patentierung ausgeschlossen. Damit war auch die letzte Schranke von praktischer Relevanz gefallen. Es gibt inzwischen fast keine Idee mehr, die in den U.S.A. nicht patentierbar wäre, wenn sie denn irgendeine wirtschaftliche Verwertbarkeit aufweist – jedenfalls im Bereich der „Software-implementierten Erfindungen“¹¹. Die Situation in der EU stellt sich noch etwas anders dar, wie wir in Kürze sehen werden.¹² Zuvor soll jedoch noch auf einen interessanten europäischen Fall hingewiesen werden.

IBM unternahm 1997, anscheinend in Absprache mit dem Europäischen Patentamt (EPA) in München, einen Vorstoß, um den Ausschluss (von der Patentierbarkeit) von Computerprogrammen „als solchen“ – so heißt es in Artikel 52 des Europäischen Patentübereinkommens (EPÜ) – zu unterhöhlen. Zu diesem Zwecke wurden zwei Patente angemeldet, die sich unmittelbar auf Programme richteten. Das EPA lehnte die Anmeldungen ab, wogegen IBM Einspruch vor der Berufungsinstanz einlegte (T935/97 und T1173/97). Nach den Anhörungen wurde 1999 entschieden, dass Software dann im Prinzip nicht mehr als Software „als solche“ zu betrachten – und somit als unpatentierbar – sei, wenn sie einen Effekt „beyond the 'normal' physical interactions between the program (software) and the computer (hardware)“ entfalten würde. Im Klartext: Innovative Software ist nie Software „als solche“, wenn sie denn überhaupt funktioniert.

In Anbetracht dieser Entscheidung wäre es nur allzu verständlich gewesen, wenn der entsprechende Passus aus dem EPÜ gestrichen worden wäre. Die Konferenz, auf der das geschehen sollte, war für Ende 2000 anberaumt. Zum Entsetzen der versammelten Fachgemeinde der Patentjuristen intervenierte jedoch die damalige Bundesjustizministerin Hertha Däubler-Gmelin unter Verweis auf nationale Sicherheitsinteressen¹³ und drohte die deutsche Delegation von der Konferenz zurückzuziehen. Das zeigte Wirkung und die Streichung von Software „als solcher“ aus der Liste der unpatentierbaren Gegenstände in Art. 52 EPÜ unterblieb.

Aufgeschreckt von den sich verschärfenden Debatten um das Pro und Kontra von Softwarepatenten sowie ermuntert durch intensive Lobbyarbeit der Stakeholder (besonders auch aus den U.S.A.¹⁴) erschien der EU Kommission Handeln geboten. Unter dem Titel *Die Patentierbarkeit computer-implementierter Erfindungen* wurde im Oktober 2000 ein Sondierungspapier der Generaldirektion Binnenmarkt an die Mitgliedstaaten herausgegeben. Dort heißt es zur Begründung:

„Die einzelstaatlichen Patentgesetze müssen [...] harmonisiert werden. Dies dürfte mehr Transparenz für die europäischen Unternehmen, insbesondere für den Mittelstand, schaffen. Ferner sollte dadurch die Wettbewerbsposition der europäischen Softwareindustrie gegenüber den Haupthandelspartnern gestärkt werden. Eine Steigerung der Wettbewerbsfähigkeit wird umso dringlicher, als die Verbreitung von Computerprogrammen über das Internet weltweit zunimmt.“ (S.2)

Ausdrücklich wollte die EU-Kommission den durch die Rechtsprechung geschaffenen *Status Quo* festschreiben (S.3), wobei man sich vorrangig auf die Entscheidungen des EPA bezog.¹⁵ Wie die angestrebte Harmonisierung denn tatsächlich die Position des Mittelstandes gegenüber den Haupthandelspartnern – gemeint sind die U.S.A. und Japan – stärken würde, wurde allerdings nicht weiter ausgeführt.

Die Reaktionen auf das Sondierungspapier fielen unterschiedlich aus. Große Patentanmelder und Patentjuristen begrüßten den Vorstoß, während sich gleichzeitig eine energische Gegenlobby herausbildete.¹⁶ Während die einen das geneigte Ohr der EU-Kommission hatten, erreichten die anderen eine nicht unwesentliche Öffentlichkeitswirkung. Die EU-Kommission sah sich 2001 schließlich veranlasst, in einer offenen Umfrage die Argumente beider Seiten einzuholen.¹⁷

Hätte man die nackten Zahlen für sich sprechen lassen, so wäre der Kommission nicht viel zu tun geblieben: Über 90% der Eingaben sprachen sich gegen die offizielle Einführung von Softwarepatenten aus. Mit diesem unbehaglichen Resultat wollte man sich in Brüssel augenscheinlich nicht zufriedengeben. Hilfe kam von der mit der Umfrage beauftragten Agentur in London. Nach einem tiefen Blick in die Zahlen gelangte man dort zu der „weisen“ Einsicht, dass die wirtschaftliche Mehrheit ja eigentlich doch für Softwarepatente sei, auch wenn sie für nicht einmal 10% der Meinungsäußerungen verantwortlich sei. Die Botschaft kam in Brüssel an, und im Februar 2002 wurde der Vorschlag für eine Richtlinie zur Patentierbarkeit Computer-implementierter Erfindungen vorgelegt. Er folgte im Wesentlichen den vom

Sondierungspapier vorgezeichneten Linien, d.h. der *Status Quo* der EPA-Praxis soll festgeschrieben werden.

Der Richtlinienentwurf jedoch erreicht die Klarstellung nicht, über deren Notwendigkeit sich Gegner und Befürworter der Patentierbarkeit von Software einigermaßen einig sind. Die EU-Kommission will den Wettbewerb nicht über Gebühr behindern und spricht sich im Prinzip gegen Patente auf Geschäftsmethoden aus. Auf der anderen Seite sind die vorgeschlagenen Formulierungen nicht wirklich dazu geeignet ihnen in der Praxis einen Riegel vorzuschieben. Was denn die in der Richtlinie vorgeschlagene „technical contribution“ leisten würde, um Patente auf Geschäftsmethoden künftig zu verhindern, wusste die zuständige Berichterstatterin des Europaparlaments den Abgeordneten auf Nachfragen hin nicht zu erklären, wehrte sich aber dagegen eine präzisere Formulierung zu finden.

Das 21. Jahrhundert – Software-Rechtsschutz, quo vadis?

Wie sieht die IT-Perspektive des 21. Jahrhunderts aus?

- Der weitgehende Zusammenbruch der „new economy“ bildete den Auftakt des 21. Jahrhunderts, jedenfalls was die Informationswirtschaft anbelangt. Auch IT muss sich hinfort rechnen.
- Unter dem Schlagwort vom „pervasive computing“ wird alles mit allem und jede mit jedem vernetzt, das ganze möglichst mobil und drahtlos.
- „Content is king“ und würde ohne maximalen Schutz durch Gesetz, Vertrag und Code (L. Lessig) erst gar nicht in die Netzwerke eingespeist, die dann ein trostloses Dasein fristen müssten – so jedenfalls die Position der Content-Industrie. Derweil finden P2P-Anhänger innovative Nutzungsformen für ihre Breitbandzugänge, die von deren Erfindern so nicht geplant waren.
- Ein vor nicht einmal dreißig Jahren gegründetes U.S.-Softwareunternehmen, Microsoft, ist zum reichsten Unternehmen der Welt aufgestiegen; sein oberster Software Engineer Bill Gates zum reichsten Mann der Welt.
- Begleitet von einem gewissen ideologischen Getöse auf Seiten sowohl der Pro- als auch der Antagonisten – die ACM-Zeitschrift Queue spricht gar von religiösen Kriegen¹⁸ – findet im Quelltext vertriebene Software mit sehr liberalen Lizenzbedingungen den Weg auf PCs, in Mobiltelefone und Automobilsteuerungen.
- Bioinformatik, Grid-Computing ... usw. usf.

Die IT-Landschaft wird vielfältiger und somit unübersichtlicher, während Software zugleich als Kohäsionskraft gefordert ist. Ohne Software liefe im wahrsten Sinne nichts mehr und die Verteilungskämpfe um die Softwaremärkte der Zukunft sind in vollem Gange. Was die Wahl der Waffen angeht, scheinen Patente eine nachgerade magische Anziehungskraft zu entfalten; einige sehen sie gar als die »‘intelligenten Bomben‘ in den Geschäftskriegen von morgen« an (Rivette & Kline 2000,

S. 40). Die Patentämter in aller Welt liefern permanent neue „Munition“ (und sind dabei z.T. recht erfinderisch), wobei sie von Gerichten durchaus unterstützt werden.

In der Debatte um Modifikationen am Richtlinienentwurf der EU-Kommission ist man dem ursprünglichen Geist zum Zeitpunkt der Festlegung, dass Computerprogramme „als solche“ nicht patentierbar seien, wieder näher gerückt. Viele Vorschläge der Kritiker – vorrangig kleine und mittlere Unternehmen, Wissenschaftler, Gewerkschaftler und Engagierte aus dem Free Software-/Open Source-Umfeld – wurden gehört und zum Teil in Modifikationen des Parlaments berücksichtigt. EU-Binnenmarktkommissar Frits Bolkestein allerdings „was not amused“ (ebenso Berichterstatterin Arlene McCarthy) und hat gedroht die Änderungen im EU-Rat rückgängig machen zu lassen. Sollten sich EU-Parlament und EU-Kommission im Kodifizierungsverfahren nicht auf einen Richtlinienentwurf einigen können, würde im Prinzip genau das erreicht werden, was die EU-Kommission ja als ihr ursprüngliches Ziel angegeben hatte: Festschreibung (durch Unterlassung) des *Status Quo*. Angesichts der fortschreitenden Praxis von EPA und nationalen Patentämtern sowie der Rechtsprechung der Gerichte in den EU-Mitgliedsstaaten wäre damit allerdings niemandem langfristig geholfen.

Der Streit um Software-Patente wäre wohl nie in der zu beobachtenden Heftigkeit entbrannt, wenn es mit Free Software/Open Source Software nicht ein Modell zur Bereitstellung dringend benötigter Güter (Software) gäbe, das so gar nicht in die vertrauten Kategorien passen will. Die in den Lehrbüchern gepredigte Rationalität des Patentwesens – Belohnung des genialen Erfinders; Förderung von Innovation durch Schaffen von monetären und ideellen Anreizen; ohne strikten Schutz geistigen Eigentums kein Fortschritt, sondern Verfall – wird mit einem Male nicht mehr einfach hingenommen, sondern erneut hinterfragt: Ist geistiges Eigentum wirklich so notwendig, wie diejenigen gebetmühlenartig wiederholen, die davon profitieren? Ist der absolute Schutz von Erfindungen im Bereich der Software tatsächlich dem Gemeinwohl zuträglich oder nicht vielleicht doch in erster Linie Fortschritts- und Wettbewerbshemmnis, wie etliche Wissenschaftler¹⁹ argwöhnen?

Empirische (Er-)Kenntnisse sind rar gesät. Es fehlt an Evidenz für die Position der Befürworter eines weitgehenden Patentschutzes fast ebenso wie an Belegen für die Gegenposition. Die streckenweise absurd anmutende Verbalakrobatik in der so genannten „Technizitätsdebatte“ mag als Beispiel dafür dienen, wie man sich im Dickicht der Hermeneutik verlieren kann, ohne überzeugende Politikziele zu formulieren.²⁰

Wohin auch immer die Entwicklung gehen mag, dass wir an der Schwelle zum Zeitalter einer „Lizenzökonomie“ stehen, ist kaum mehr zu übersehen. Der verfassungsrechtlich verbrieften, wirtschaftlichen Freiheit des Individuums wird, wenn es denn seine Freiheit nutzen will, die Unumgänglichkeit von Lizenzverhandlungen zur Seite gestellt. Innovation ohne Verhandlung wird in Zukunft nicht mehr die Regel sondern die Ausnahme sein und jedes Originalgenie ist gut beraten sich bei Zeiten einen Anwalt zu suchen – oder vielleicht selbst besser Anwalt zu werden. Das könnte lohnenswerter sein, denn schließlich wachsen die in Prozessen erstrittenen Schadensersatzzahlungen um ein Mehrfaches schneller als das Bruttoinlandsprodukt, jedenfalls in den U.S.A. (Kielholz 2003). Und wie wir gesehen haben, sind

die U.S.A. oft genug der Vorreiter, dem die anderen Länder dann folgen. ...

- 1 Der Aufsatz beruht in Teilen auf Kretschmer (2003).
- 2 Dazu Bernhard Dotzler, Hrsg. (1996): *Babbages Rechen-Automate*, Springer, Wien und New York.
- 3 Nachdem die geschäftlich erfolglose Eckert/Mauchly-Firma von Remington-Rand übernommen und später mit Sperry Gyroscope fusioniert war, prozessierte sie bis in die 70er hinein gegen andere Computerhersteller wegen Patentverletzung. Das 1973 gefällte abschließende Urteil fiel allerdings zu ihren Ungunsten aus. (Macrae 1994, S. 256)
- 4 Von Neumann (1945).
- 5 Das Paradigma der freien Softwareentwicklung könnte demnach alles andere als zufällig sein.
- 6 Die historischen Ausführungen zur Softwareindustrie in diesem Abschnitt beruhen zum großen Teil auf dem Aufsatz von Johnson (2002).
- 7 Das erste derartige Unternehmen, Computer Usage Corporation, wurde bereits 1955 gegründet (Johnson 2002, S.14).
- 8 House Report, 94th Congress, 1476 (1975).
- 9 Ähnliche Argumente gab es einige Jahre später auch von den Protagonisten freier Software um Richard Stallman zu hören, allerdings unter Ablehnung sowohl des Patent- als auch des Copyright-Schutzes von Software.
- 10 Vgl. etwa Parker v. Flook (437 U.S. 584, 1978), wo das Gericht befand: »The process itself, not merely the mathematical algorithm, must be new and useful« (S.591).
- 11 So die EU-Terminologie für Softwarepatente.
- 12 Der wesentliche Unterschied besteht hinsichtlich der Patentierbarkeit von reinen Geschäftsmethoden (computer-implemented business methods), die in Europa i.d.R. nicht ohne weiteres patentierbar sind. Stattdessen müssen die Ansprüche die beanspruchten Methoden als komplexe Rekonfigurierung des Computers präsentieren. Insgesamt sind die Unterschiede hinsichtlich der Patentierbarkeit marginal, wie Beresford (2000) gezeigt hat.
- 13 Zum Zusammenhang zwischen IT-Sicherheit und Patentschutz für Software siehe Gehring (2003).
- 14 Das dürfte nicht verwundern, wird der größte Anteil an europäischen Softwarepatenten doch von U.S.-Unternehmen bzw. Töchtern von U.S.-Unternehmen gehalten; japanische Unternehmen folgen auf Platz 2. Die stark von kleinen und mittleren Unternehmen geprägte europäische IT-Landschaft setzt traditionell nicht so sehr auf Softwarepatente und landet deshalb recht abgeschlagen auf Platz 3.
- 15 Das Europäische Patentamt ist bemerkenswerterweise keine EU-Institution. Seine Patentierungspraxis und Rechtsprechung fallen wesentlich häufiger zugunsten der Patentanmelder aus als etwa die der nationalen Patentämter und -gerichte in den EU-Mitgliedsstaaten. Die Heranziehung der EPA-Praxis als Maßstab für die Feststellung des Status Quo seitens der EU-Kommission spricht insofern für deren Bestreben eine möglichst weitgehende Patentierbarkeit von Software erreichen zu wollen.
- 16 Siehe z.B. die Eurolinux-Initiative: <http://www.eurolinux.org/>.
- 17 Die EU-Studie: http://www.europa.eu.int/comm/internal_market/de/indprop/comp/softpatanalyse.htm.
- 18 Vgl. ACM Queue Juli/August 2003.
- 19 Eine Auswahl an aktuelleren Aufsätzen: Heller (1998), Heller & Eisenberg (1998), Long (2000), Bessen & Maskin (2000), Burk (2000), Gallini & Scotchmer (2001), Bagley (2001), Pénin (2003) und Hall (2003).
- 20 Während es unter Technikphilosophen und Anthropologen unumstritten ist (vgl. z.B. Rammert 1999), dass nicht nur menschliche Artefakte, sondern auch Handlungsweisen dem Technikbegriff – nicht zu verwechseln

mit dem Technologiebegriff – unterliegen, schaffen es Patentfachleute seit Jahrzehnten, sich über die Technizität von Software zu streiten. Doch darauf kommt es gar nicht an. Wo Patentschutz dem Gemeinwohl abträglich ist, Technik hin oder her, sollte er nicht gewährt oder geeignet eingeschränkt werden.

Literatur

- M.A. Bagley (2001): *Internet Business Model Patents: Obvious By Analogy*, 7 Mich. Telecomm. Tech. L. Rev. 253 (2001), <<http://www.mttl.org/volveven/bagley.html>>.
- K. Beresford (2000): *Patenting Software under the European Patent Convention*, Sweet & Maxwell, London.
- J. Bessen & E. Maskin (2000): *Sequential Innovation, Patents, and Imitation*, Working Paper No. 00-01, MIT, Department of Economics, Cambridge, MA.
- D.L. Burk (2000): *Patenting Speech*, 79 Texas L. Rev. 99 (2000), <<http://www.isc.umn.edu/research/papers/Patent2.pdf>>.
- A.W. Branscomb (1994): *Who Owns Information?* Basic Books, New York, NY
- N. Gallini & S. Scotchmer (2001): *Intellectual Property: When is it the best incentive system?* (August 1, 2001). Economics Department, University of California, Berkeley, Working Paper E01-303, <<http://repositories.cdlib.org/iber/econ/E01-303>>.
- R.A. Gehring (2003): *Software Development, Intellectual Property, and IT Security*, 2003 (1) *The Journal of Information, Law and Technology (JILT)*, <<http://elj.warwick.ac.uk/jilt/03-1/gehring.html>>.
- B.H. Hall (2003): *Business Method Patents, Innovation, and Policy*, Working Paper No. 9717, National Bureau of Economic Research, Cambridge, MA.
- M.A. Heller (1998): *The Tragedy of the Anticommons: Property in the Transition from Marx to Markets*, *Harvard Law Review*, Vol. 111, S. 623-688.
- M.A. Heller & R.S. Eisenberg (1998): *Can Patents Deter Innovation? The Anticommons in Biomedical Research*, *Science*, Vol. 280, 1998, S. 698-701.
- L. Johnson (2002): *Creating the Software Industry*, in: *IEEE Annals of the History of Computing*, Vol. 24, No. 1, 14-42.
- C.V. Jones (1968): *Software is unpatentable*, *New York Times* v. 23.10.1968, S. 59.
- W.B. Kielholz (2003): *Prozesslawinen bedrohen weltweit die Versicherbarkeit*, in: *Neue Zürcher Zeitung* v. 4./5. Oktober 2003, S. 19.
- M. Kretschmer (2003): *Software as Text and Machine: The Legal Capture of Digital Innovation*, 2003 (1) *The Journal of Information, Law and Technology (JILT)*, <<http://elj.warwick.ac.uk/jilt/03-1/kretschmer.html>>.
- C. Long (2000): *Patents and Cumulative Innovation*, *Washington University Journal of Law & Policy*, Vol. 229, S. 229-246.
- N. Macrae (1994): *John von Neumann*, Birkhäuser, Basel, Boston und Berlin.
- J. von Neumann (1945): *First Draft of a Report on the EDVAC*, reprinted in: *IEEE Annals of the History of Computing*, October-December 1993, Vol. 15, No. 4, S. 27-75; TIFF-Bilder des Originals online unter: <<http://www.wps.com/projects/EDVAC/>>.
- J. Pénin (2003): *Patents versus ex-post rewards: a new look*, Working Paper No. 20-19, University of Québec, Economics Department, Québec.
- W. Rammert (1999): *Technik*, in: H.J. Sandkühler, Hrsg.: *Enzyklopädie Philosophie*, Felix Meiner, Hamburg, S. 1602-1613.
- K.G. Rivette, D. Kline (2000): *Wie sich aus Patenten mehr herausholen lässt*, in: *Harvard Business Manager* 4/2000, S. 28-40.