

Patentschutz und Softwareentwicklung – ein unüberbrückbarer Gegensatz?

ANDREAS WIEBE

I. Einleitung

Fast alle Schutzsysteme des Immaterialgüterrechts lassen sich für den Schutz von Software nutzbar machen. Bereits seit Beginn der Verbreitung der Computertechnik standen das Urheberrecht und das Patentrecht im Vordergrund. Dabei zeichnete sich schnell ein Trend zum Urheberrecht als primärem Schutzrecht ab, der durch die Richtlinie zum Urheberrecht von 1991 und die darauf folgende Harmonisierung in Europa verfestigt wurde.¹ In den neunziger Jahren wurde in den USA neben dem Know-how-Schutz auch verstärkt der Patentschutz für Software genutzt, was vor allem im Bereich geschäftlicher Anwendungen zu einigen sehr breiten Patenterteilungen führte.

Dies hat, vor allem auf der Seite der Open-Source-Bewegung, die Befürchtung geweckt, dass auch in Europa eine breite Patentierung von Software erfolgen könnte. Wegen der gegenüber dem Urheberrecht andersartigen Ausgestaltung hinsichtlich Schutzgegenstand und -umfang ist es nicht nur zweifelhaft, ob sich der Patentschutz in gleicher Weise für das Open-Source-Modell nutzbar machen lässt, sondern es stellt sich darüber hinaus die Frage, ob nicht die Ausweitung des Patentschutzes auf Computerprogramme Open Source als Entwicklungs- und Vertriebsmodell grundsätzlich ebenso wie im Einzelfall unmöglich macht.

II. Software als technisches Produkt

1. Patentrecht als Alternative zum Urheberrechtsschutz

Nun lässt sich nicht von vornherein ausschließen, dass beide Schutzsysteme eine „friedliche Koexistenz“ führen können. Die bisherigen Abgrenzungsversuche konzentrierten sich in Deutschland und Europa auf das Erfordernis der Technizität. Das Problem liegt jedoch darin, dass Software eine „Doppelnatur“ aufweist.² Der Entwicklungsprozess von Software beinhaltet die Entwicklung einer Lösung für ein bestimmtes Problem mit zunehmendem Konkretisierungsgrad. Dabei entstehen verschiedene „Zwischenprodukte“ bis hin zum lauffähigen „Endprodukt“, dem Objektformat. Soweit dabei in einer Programmiersprache erstellte Konstrukte betroffen sind, sind diese als Ausdrucksformen geistigen Schaffens dem Urheberrecht zuzuordnen. Soweit es jedoch um die Funktionalität des „Running Code“ geht, also

¹ Richtlinie 91/250/EWG vom 14.5.1991, ABIEG L 122 vom 17.5.1991, 42; dazu Blocher/Walter, in: Walter (Hrsg.), Europäisches Urheberrecht, Wien/New York 2001, S. 111 ff.

² Vgl. bereits Wiebe, Know-how-Schutz von Computersoftware, München 1993, S. 424 ff.; zuletzt im vertragsrechtlichen Kontext Hilty, MMR 2003, 3, 8 ff.

seine „Bestimmung“, im Zusammenwirken mit einer Hardware bestimmte Aufgaben zu bewältigen, lässt sich dieser Aspekt dem Patentrecht zuordnen.³

Dies lässt sich etwas vereinfacht an dem Unterschied von Quellformat und Objektformat zeigen. Im Quellformat weist das Computerprogramm als Darstellungsform einen in einer Programmiersprache verfassten Text auf. Dieser linguistische Aspekt spiegelt sich in der Definition von DIN 44300 wieder:

Eine zur Lösung einer Aufgabe vollständige Anweisung (d.h. eine in einer beliebigen Sprache abgefasste Arbeitsvorschrift, die im gegebenen Zusammenhang wie auch im Sinne der benutzten Sprache abgeschlossen ist) zusammen mit allen erforderlichen Vereinbarungen (d.h. Absprachen über in Anweisungen auftretende Sprachelemente).⁴

Im Objektformat dagegen handelt es sich um die maschinenlesbare Fassung desselben immateriellen Guts Computerprogramm, bei dem dann der technisch-funktionelle Aspekt im Vordergrund steht:

Eine Folge von Befehlen, die nach Aufnahme in einen maschinenlesbaren Träger fähig sind zu bewirken, dass eine Maschine mit informationsverarbeitenden Fähigkeiten eine bestimmte Funktion oder Aufgabe oder ein bestimmtes Ergebnis anzeigt, ausführt oder erzielt.⁵

Wo soll nun der Schwerpunkt des Schutzes liegen? Geht es um die Förderung kreativer Tätigkeit oder um die Förderung technischer Innovation? Eigentlich liegt jeder herkömmlichen Maschine ein Algorithmus zu Grunde, der durch Interaktion mechanischer oder elektronischer Bauteile ausgeführt wird. Die Computertechnik ersetzt diese Art der Implementierung durch Arbeitsanweisungen an einen Universalcomputer. Im Mittelpunkt der zweiten industriellen Revolution, dem Informationszeitalter, steht aber auch die Ersetzung geistiger Tätigkeiten durch computergestützte Verfahren.

Der Universalrechner ist keine klassische Maschine mehr, sondern erst die Software bestimmt die Funktionalität. Sie ist eine „transklassische Maschine“, oder anders ausgedrückt: „Der Algorithmus ist die Maschine“.⁶ Damit wird es möglich, Materie und Energie durch Information zu ersetzen. Folge ist, dass ein immer größerer Teil der Innovationstätigkeit sich auf Softwareerstellung verlagert. Dies spricht eher dafür, Software als technisches Produkt anzusehen, das in den Bereich des Patentrechts gehört.⁷

Dann ist davon auszugehen, dass sich Software nicht so fundamental von anderen technischen Gegenständen unterscheidet, dass auch dafür die Prämisse der För

³ Eingehend dazu und mit dem Versuch einer „Synthese“ Horns, GRUR 2001, 1, 2 ff.

⁴ DIN 44300 Teil 4 Nr. 4.1.9 (1988), zit. nach Haberstumpf, in: Lehmann (Hrsg.), Rechtsschutz und Verwertung von Computerprogrammen, 2. Aufl., Köln 1993, S. 78.

⁵ § 1 Mustervorschriften WIPO, GRUR Int. 1978, 286.

⁶ Troller, CR 1987, 278, 280.

⁷ Vgl. bereits Dryja, Looking to the Changing Nature of Software for Clues to its Protection, University of Baltimore Intellectual Property Law Journal, vol. 3, 1995, No. 2, S. 109 ff., der darauf hinweist, dass mit dem Übergang zur objektorientierten Programmierung der Programmierer sich nicht mehr als „artist“, sondern als „engineer“ verstehe.

derung innovativer Entwicklungen gelten muss.⁸ Allerdings gibt es in der ökonomischen Forschung keine schlüssige Antwort auf die Frage nach der Sinnhaftigkeit des Patentrechts.⁹

Einige Vorteile der Patentierung kann man hier bereits nennen. Das Patentrecht ist in vielfältiger Weise auf technische Erfindungen zugeschnitten.¹⁰ Anders als das Urheberrecht schützt das Patentrecht das eigentlich Wertvolle bei Computerprogrammen, nämlich die Algorithmen.¹¹ Es ist ein effektiv auf die Technik zugeschnittenes, starkes, aber dafür zeitlich gegenüber dem Urheberrecht stärker begrenztes Schutzrecht. Es hat, und darin trifft es sich mit dem Anliegen der Open-Source-Bewegung, auch den Zweck, technisch innovative Ideen frühzeitig offen zu legen (Informationsfunktion) und damit das allgemeine Wissen zu bereichern. Es regt insoweit die innovative Tätigkeit an, um das Patent „herum“ zu erfinden, also „Umgehungserfindungen“ zu tätigen und damit den Substitutionswettbewerb zu fördern. Das Patent begründet keine Monopolstellung, sondern dient der Förderung des Wettbewerbs. Hinzu kommt: Patentschutz ist auch ein Schutzinstrument für kleine und mittlere Unternehmen (KMU) gegen große Unternehmen. Vielleicht wäre Microsoft mit umfassender Softwarepatentierung nicht so groß geworden, da die kleinen Mitbewerber ihre Ideen besser hätten schützen können.

Allerdings muss bereits an dieser Stelle gesagt werden, dass es stark darauf ankommt, dass die Patentansprüche nicht zu breit gefasst und nicht zu nachlässig geprüft werden, damit ein Patent nicht innovationshindernd wirkt. Ein weiterer Nachteil kommt hinzu: Wegen der hohen Schutzvoraussetzungen des Patentrechts ist nur ein kleiner Teil der Programme geschützt, also die wirklich innovativen. Daneben besteht aber auch ein Bedürfnis nach einem breiten Schutz gegen Piraterie. Dieses kann sowohl durch einen sehr eng verstandenen Urheberrechtsschutz als auch durch einen ergänzenden wettbewerbsrechtlichen Leistungsschutz gewährleistet werden.¹²

⁸ Vgl. demgegenüber Lutterbeck/Gehring/Horns, Sicherheit in der Informationstechnologie und Patentschutz für Software-Produkte – Ein Widerspruch?, Kurzgutachten, Berlin 2000, <http://www.sicherheit-im-internet.de/download/Kurzgutachten-Software-patente.pdf>, S. 59 f., mit der gegenteiligen Schlußfolgerung unter Berufung auf ökonomische Studien.

⁹ Vgl. Blind/Edler/Nack/Straus, Softwarepatente. Eine empirische Analyse aus ökonomischer und juristischer Perspektive, Heidelberg 2003, S. 103, 204; Ullrich, Grenzen des Rechtsschutzes: Technologieschutz zwischen Wettbewerbs- und Industriepolitik, in: Schrickler/Dreier/Kur (Hrsg.), Geistiges Eigentum im Dienst der Innovation, Baden-Baden 2001, S. 83, 100 ff. m.w.Nachw.; Grandstrand, The Economics and Management of Intellectual Property, Cheltenham, UK, Northampton, USA 2000; Lutterbeck/Gehring/Horns, S. 50 ff.

¹⁰ Vgl. auch die Diskussion der Vor- und Nachteile des Patentschutzes für Software bei Haase, Die Patentierbarkeit von Computerprogrammen, Hamburg 2003, S. 176 ff.

¹¹ Diese Aussage ist allerdings abhängig von der zu Grunde gelegten Definition von Algorithmen, vgl. Wiebe, in: Spindler (Hrsg.), Rechtsfragen der Open Source Software, Köln 2004, Kap. F Rn. 40 f.; Horns, GRUR 2001 1, 5 ff.

¹² Vgl. bereits Wiebe, Know-how-Schutz von Computersoftware, München 1993, S. 424 ff.

2. Geschichte der Softwarepatentierung: Ein langer Leidensweg

Die Frage, ob es sich bei softwarebezogenen Erfindungen um technische Gegenstände handelt, hat die Rechtsprechung seit 30 Jahren beschäftigt.¹³ Dazu ein kleines Einführungsbeispiel.

a) Beispielfall: „Automatische Absatzsteuerung“

Das BPatG hatte im Jahre 2000 über folgende Patentanmeldung zu entscheiden: „Verfahren zur automatischen Absatzsteuerung für eine Menge von Waren und/oder Dienstleistungen unter Verwendung eines digitalen Verarbeitungssystems, [...] gekennzeichnet durch folgende Verfahrensschritte:

- elektronisches Speichern von zeitlichen Absatzprognosedaten für mindestens einen bestimmten Abgabepreis der Waren/Dienstleistungen;
- elektronisches Erfassen aktueller Absatzdaten für die Menge von Waren/Dienstleistungen;
- elektronische Auswahl eines angepassten Abgabepreises als Funktion der Abweichung der aktuellen Absatzdaten von den Absatzprognosedaten;
- Anzeigen des ausgewählten Abgabepreises.“¹⁴

Vereinfacht ging es dabei um einen Getränke-Automaten, bei dem programmgesteuert der Preis der Getränkedosen an die elektronisch erfasste Nachfrage angepasst wurde. Dieser Anspruch ist paradigmatisch für die technische Entwicklung. Was seit Jahrhunderten der Kaufmann hinter dem Ladentisch mit Bleistift und Kontobüchern machte, wird heute durch ein Computerprogramm übernommen und vollautomatisch durchgeführt. Das Besondere an dem Anspruch ist aber, dass nicht nur der Absatzvorgang selbst, sondern auch die betriebswirtschaftliche Regel automatisiert wird, wonach der Marktpreis sich nach Angebot und Nachfrage regelt.

b) Technische Entwicklung und Zweck des Patentrechts

Eine Erfindung als Gegenstand des Patentrechts ist eine „Lehre zum planmäßigen Einsatz beherrschbarer Naturkräfte zur Erreichung eines kausal übersehbaren Erfolges.“¹⁵ Mit dem Bezug auf den Einsatz beherrschbarer Naturkräfte ist zugleich das Technikerfordernis Teil des Erfindungsbegriffs. Eigentlich stammt das Technikerfordernis bereits aus einer Entscheidung des Reichsgerichts (RG) des Jahres 1933, in der es um die Patentfähigkeit einer Multiplikations- und Divisionstabelle in Buchform ging (RG GRUR 1933, 289).

Eine Erfindung müsse „ihrem Wesen nach“ technisch sein, also mit den Mitteln von Naturkräften arbeiten, während das Rechenwerk von Tabellen dem Gebiet rein geistiger Tätigkeit angehöre und außer Betracht bleiben müsse. „Die Technik bezieht sich auf die Erscheinungswelt im Gegensatz zur Welt des Geistes; sie arbeitet nach den Lehren von Physik und Chemie“ (RG GRUR 1933, 289, 290).

¹³ Eine Übersicht zur Entwicklung des Patentschutzes von Software geben Tauchert, GRUR 1999, 829 ff.; Nack, GRUR Int. 2000, 853 ff.; Kraßer, GRUR 2001, 959 ff.; Schölch, GRUR 2001, 16 f.; Anders, GRUR 2001, 555 ff.; Klopmeier, Mitt. 2002, 65 ff.; Sedlmaier, Mitt 2002, 55 ff.

¹⁴ BPatG, CR 2000 97.

¹⁵ BGH GRUR 1969, 672 – Rote Taube.

Das Patentrecht ist ein Kind der Industrialisierung. Sein Ziel war es von Anfang an, die Innovationstätigkeit zu fördern und damit technischen Fortschritt anzulegen. Technischer Fortschritt bedeutete Fortschritt der gewerblichen Güterproduktion und wirtschaftliches Wachstum. Das hieß im 19. Jahrhundert: Entwicklung neuer Maschinen und Ersetzung physischer Tätigkeit. Der Einsatz von Naturkräften, Materie und Energie, führt zu Einsparung von Arbeitskraft und/oder Kapital.

Der Siegeszug des Computers führte zur Automatisierung sowohl von Maschinen- als auch von Geistestätigkeit. Jeder herkömmlichen Maschine und jedem technischen Verfahren liegt ein Algorithmus zu Grunde, der durch Interaktion mechanischer oder elektronischer Bauteile ausgeführt wird. Die Computertechnik ersetzt diese Art der Implementierung durch Arbeitsanweisungen an einen Universalcomputer. „Der Algorithmus ist die Maschine.“ Im Mittelpunkt der zweiten industriellen Revolution, dem Informationszeitalter, steht aber auch die Ersetzung geistiger Tätigkeiten durch computergestützte Verfahren. Die Rechtsprechung stand daher vor der Frage, ob solche Verfahren zum Gebiet der Technik gehören und damit patentfähig sein können.

Diese Frage wirkt zunächst überraschend: ein Computer ist ein technisches Gerät und die Software steuert dieses technische Gerät: Software und Hardware gehören untrennbar zusammen. Wie kann es also Zweifel geben, dass ein softwaregestütztes Verfahren technischen Charakter hat?

Ein Beispiel soll die Problematik noch deutlicher machen. Jemand konzipiert eine Software zur Berechnung von Versicherungsprämien. Im Mittelpunkt steht dabei eine neuartige mathematische Formel, die das tatsächliche Versicherungsrisiko besonders präzise erfasst und daher sehr günstige Versicherungsprämien ermöglicht.¹⁶ Diese Software lässt sich auf einem handelsüblichen PC ausführen. Ist das Computerprogramm patentfähig?

Dabei sind mehrere Aspekte auseinander zu halten. Der Computer ist ein technisches Gerät. Software bedarf des Computers zur Ausführung ihrer Funktionen, beide sind also nur zusammen sinnvoll einsetzbar. Das eigentlich Innovative bei diesem Fall ist die Entwicklung der Formel, deren Gegenstand nicht zum traditionellen Gebiet der Technik gehört. Deren Implementierung in eine Software ist wiederum eine handwerkliche Leistung, die von jedem Programmierer durchgeführt werden kann. Durch die Implementierung in Software wird ein traditionell nichttechnischer Gegenstand in ein „Gewand“ gekleidet, das zum Bereich der Technik gehört.

In der über dreißigjährigen Geschichte der Softwarepatentierung ist eine Brandmauer nach der anderen gegen die Patentierung solcher Ansprüche gefallen. Die Entwicklung kann hier nicht nachgezeichnet werden.¹⁷ Es bleibt jedoch festzuhalten, dass vor allem die Rechtsprechung der deutschen Gerichte, zu einem geringeren Grad auch die des EPA, durch starke Einzelfallbezogenheit und Mangel an klaren und verständlichen Regeln gekennzeichnet war. Insgesamt ist jedoch ein klarer Trend von einer auf technische Steuerprogramme und Betriebssysteme beschränkten Patentierung hin zu einer Auffassung, die nahezu jede Softwareimplementierung als technisch ansieht, zu konstatieren. Der BGH hat zuletzt festgestellt, dass, wenn

¹⁶ Vgl. Nack, GRUR Int. 2000, 853; EPA, CRi 2001, 18.

¹⁷ S. die Nachw. o. Fußn. 13.

die industrielle Entwicklungstätigkeit zunehmend automatisiert durchgeführt würde, auch der Technikbegriff angepasst werden müsse. Entsprechend könne auch auf das Erfordernis des unmittelbaren Einsatzes beherrschbarer Naturkräfte verzichtet werden.¹⁸

Während der Ausschlussstatbestand für Computerprogramme als solche nach § 1 Abs. 2 Nr. 3, Abs. 3 PatG unklar ist und kaum praktische Bedeutung erlangt hat,¹⁹ wirft dies vor allem Probleme hinsichtlich der Abgrenzungsfunktion des Ausschlussstatbestands für Geschäftsmethoden auf. Die vom BGH getroffene – eigentlich selbstverständliche – Feststellung, dass der Technikbegriff nicht statisch sei, ist hierbei von großer Bedeutung.²⁰ Wenn denn das (im Gesetz gar nicht explizit enthaltene, aber gewohnheitsrechtlich anerkannte) Technikerfordernis eine Abgrenzungsfunktion gegenüber nicht für schutzwürdig erachteten geistigen Leistungen hat,²¹ so ist diese nach Sinn und Zweck des Patentrechts zu bestimmen. Diese muss natürlich auch Veränderungen des Normbereichs berücksichtigen. Dazu gehört nicht nur die technische Entwicklung, sondern auch die damit zusammenhängenden gesellschaftlichen Entwicklungen. So stellte der BGH in der „Rote Taube“-Entscheidung²² fest, dass die zum Erlass des PatG 1877 herrschende Auffassung über patentfähige Erfindungen nicht mehr maßgebend sein könne, da sich „inzwischen Naturwissenschaft und Technik erheblich gewandelt haben.“²³

Ein modernes Technikverständnis umfasst nicht nur die Verlagerung herkömmlicher Maschinenteknik auf computergesteuerte Verfahren, sondern auch die Verlagerung bisheriger gedanklicher Arbeit auf Computer – und damit auf technische Verfahren. Hier findet Innovation statt und ist die Quelle von Wirtschaftswachstum. Solange es sich um eine software- oder hardwarebezogene Anspruchsformulierung handelt, bewegen wir uns im Bereich der Technik.

Sorgfältig zu prüfen sind dann die Merkmale Neuheit und erfinderische Tätigkeit. Diese weiteren Anforderungen schaffen Freiräume für die ganz alltäglichen Weiterentwicklungen. In diesem Bereich sind auch die Sünden der amerikanischen Praxis angesiedelt, die häufig bei sehr breiten Ansprüchen nicht hinreichend Neuheit und erfinderische Tätigkeit gewürdigt hat. So wurde die Unterlassungsverfügung für den „one-click-process“ von Microsoft wieder aufgehoben, da bereits früher solche Verfahren genutzt worden seien.

c) Richtlinienvorschlag der EU-Kommission

Unter diesen Vorzeichen war der ursprüngliche Richtlinienvorschlag eine echte Befreiung.²⁴ Der Vorschlag ging davon aus, dass zunächst alle Computerprogramme

¹⁸ BGHZ 143, 255 – Logikverifikation.

¹⁹ Vgl. Bittner, in: Bröcker/Czychowski/Schäfer (Hrsg.), Praxishandbuch Geistiges Eigentum im Internet, München 2003, S. 679, 686; Wiebe, in: Spindler (Hrsg.), Rechtsfragen bei Open Source, Köln 2004, Kap. F, Rn. 23 ff.

²⁰ Vgl. Hössle, Der nicht-statische Technikbegriff, Mitt. 2000, 343 ff.

²¹ BGH GRUR 1992, 36 – Chinesische Schriftzeichen.

²² Im Fall „Rote Taube“ ging es um die Erweiterung des Naturkräfte-Begriffes von der Physik/Chemie auf die Biologie.

²³ BGH GRUR 1969, 672.

²⁴ Vorschlag für eine Richtlinie des Europäischen Parlaments und des Rates über die Patentierbarkeit computerimplementierter Erfindungen, KOM(2002) 92 endg. v. 20.2.2002, ABIEG C 2002/151 E

technischen Charakter aufweisen, da sie auf einem Computer ablaufen (Art. 3). Im Rahmen des Merkmals der erfinderischen Tätigkeit wurde dann aber ein „technischer Beitrag“ gefordert (Art. 4 Nr. 2). Nur dieser technische Beitrag sollte bei der Prüfung der erfinderischen Tätigkeit berücksichtigt werden. Damit würde auf der einen Seite die unglückselige Verquickung des Technikerfordernisses mit der Frage, worin der innovative Beitrag der Erfindung liegt („Kerntheorie“), aufgehoben. Zum anderen würde am Technikerfordernis festgehalten, dieses aber nicht-statisch verstanden, sodass etwa die Notwendigkeit, im Vorfeld technische Überlegungen anzustellen bereits ausreichen kann.

Zur Illustration kann noch einmal die EPA-Entscheidung „Steuerung eines Pensionssystems“ dienen.²⁵ Dort wurde die softwareimplementierte Ausführung eines Verfahrens zur Berechnung von Versicherungsbeiträgen beansprucht. Die Automatisierung stellte einen technischen Vorgang dar, war aber keine besondere erfinderische Leistung. Demgegenüber war die finanzmathematische Formel neuartig, aber beinhaltete nicht die Lösung einer technischen Aufgabe. Da im technischen Bereich keine erfinderische Leistung gegeben war, wurde die Patentfähigkeit vom EPA abgelehnt. Die EPA-Entscheidung steht insoweit im Einklang mit dem Ansatz des Richtlinienvorschlags.

Damit wären der Patentierung geschäftsbezogener Ansprüche Grenzen gesetzt, und dies war auch im ersten Entwurf die Absicht der Kommission. Nur eine technische Innovation trägt zur Patentfähigkeit bei, nicht aber eine innovative Geschäftsmethode selbst. Bei der Bestimmung der erfinderischen Tätigkeit wird auf das Wissen und Können eines Durchschnittsfachmanns in dem jeweiligen Gebiet abgestellt (§4 PatG). Fachmann ist nicht der Ökonom, sondern der Programmierer bzw. Wirtschaftsinformatiker. Dem Fachmann wird quasi fiktiv die Kenntnis der nicht-technischen Innovationen zugerechnet, sodass diese nicht mehr zur Begründung der Patentfähigkeit beitragen können. Die routinemäßige Softwareimplementierung einer guten Geschäftsidee ist für den Programmierer handwerkliche Tätigkeit. Das Patentamt muss keine Ökonomen oder Sprachwissenschaftler als Prüfer einstellen.

Diese auf den ersten Blick klare Abgrenzung ist natürlich schwierig in Grenzbereichen. Die computerimplementierte „Fassung“ einer Geschäftsidee unterscheidet sich jedoch von der herkömmlichen Ausführungsweise. Je mehr die Ausführung der Geschäftsidee auch in ihren nichttechnischen Bestandteilen durch die Computerausführung beeinflusst und bestimmt wird, desto enger ist auch der Zusammenhang zwischen Geschäftsmethode und technischer Ausführung auf dem Rechner.

Als Beispiel lässt sich der Sachverhalt der jüngsten BGH-Entscheidung „Suche fehlerhafter Zeichenketten“ anführen.²⁶ Dabei ging es um ein neuartiges Verfahren zur Fehlerkorrektur in digital gespeicherten Texten. Diese erfolgt nicht auf der Grundlage eines Vergleichs mit abgespeicherten korrekten Wörtern, wie wir es etwa vom Word-Programm kennen, sondern durch eine statistische Analyse des Textes. Bei einem fehlerfreien Wort werden einzelne Buchstaben ausgelassen oder vertauscht und dann durch Vergleich der Auftretenswahrscheinlichkeit mögliche feh-

v. 25. 6. 2002, 129. Dazu Sedlmaier, Mitt. 2002, 97 ff.; Röttinger, CR 2002, 616 ff.

²⁵ EPA, CRi 2001, 18.

²⁶ BGH GRUR Int. 2002, 323.

lerhafte Worte ermittelt. Der Vorteil eines solchen Verfahrens ist ein sehr geringer Speicherplatzbedarf. Ein solches Verfahren würde nie jemand mit Papier und Bleistift durchführen. Es ist so stark durch Einsatz der Digitaltechnik bedingt und damit verknüpft, dass es selbst als technisch anzusehen ist.²⁷

Die Geschäftsmethode wird aber durch eine Patentierung nur in ihrer von einer technischen Implementierung abhängigen Form monopolisiert. Insoweit unterscheidet sich dieser Fall von der Entwicklung einer neuartigen finanzmathematischen Methode zur Berechnung von Versicherungsprämien die nur computerimplementiert ausgeführt wird, aber auch mit Papier und Bleistift ausgeführt werden könnte. Sie unterscheidet sich auch von dem eingangs angeführten Getränkeautomaten. Bei diesem entsprach die technische Umsetzung der betriebswirtschaftlichen Regel dem „herkömmlichen Rüstzeug des Fachmanns“ und war daher nicht erfinderisch.²⁸

Der Ansatz des Richtlinienvorschlags ist nunmehr durch das Europäische Parlament, nicht zuletzt auf Druck der Open-Source-Gemeinde, fast in sein Gegenteil verkehrt worden.

d) Entschließung des Europäischen Parlaments

In der Entschließung²⁹ wird bereits in Art. 4a und Erwägungsgrund 13a abweichend vom bisherigen Erwägungsgrund 13 festgestellt, dass die Computerimplementierung allein keinen „technischen Beitrag“ leistet. Damit sollten computerimplementierte Erfindungen nicht mehr grundsätzlich als technisch angesehen werden.

Die Definition des „technischen Beitrags“ in Art. 2b) stellt klar, dass es dabei um das Technizitätserfordernis geht.³⁰ Dabei wird auch die Einbeziehung von Information in den Bereich der Naturkräfte ebenso abgelehnt, wie eine Beschränkung des Technikbegriffs auf die Beherrschung von Naturkräften zur Beherrschung der physikalischen Wirkungen der Informationsverarbeitung festgelegt wird. Der vom BGH befürworteten Weiterentwicklung des Technikbegriffs wird damit ein Riegel vorgeschoben, mit vielleicht unübersehbaren Konsequenzen für die Zukunft des Patentrechts allgemein.

Zusätzlich wird nach Art 3a die Patentierbarkeit im Hinblick auf Methode und Ergebnis auf „industrielle Anwendungen“ beschränkt, worunter nach Art. 2 bb „die automatisierte Herstellung materieller Güter“ zu verstehen ist. Zu dieser Beschränkung auf die Industrie des 19. Jahrhunderts passt auch, dass in Art.3a zusätzlich festgestellt wird, dass Innovationen im Bereich der Datenverarbeitung nicht patentierbar sind.³¹ Nicht einmal die Einsparung von Ressourcen in einem Datenver-

²⁷ So auch Anders, GRUR 2001, 559. Die gleiche Situation ergibt sich bei Reverse Auctions. Wer teilt allen Einwohnern per Brief oder Telefon seine Bereitschaft mit, für einen Produkt einen bestimmten Preis zu zahlen. Wie kann ein potenzieller Verkäufer bei einem Angebot in der Tageszeitung sicher sein, dass er auch zum Zuge kommt, vgl. Anders, a.a. O.

²⁸ Vgl. BPatG, CR 2000 97, 99.

²⁹ P5_TA-PROV(2003)0402, Legislative Entschließung des Europäischen Parlaments zu dem Vorschlag für eine Richtlinie des Europäischen Parlaments und des Rates über die Patentierbarkeit computerimplementierter Erfindungen (KOM(2002)92 – Cs-0082/2002-2002/0047(COD)).

³⁰ Vgl. auch die neue Definition „computerimplementierter Erfindungen“ in Art. 2a).

³¹ Deren Patentierbarkeit war in der bisherigen Diskussion eigentlich bereits Konsens, vgl. Ohly, CR 2001, 809, 815.

beitungssystem gilt dann als technischer Beitrag (Art. 4b). Damit geht die Entschließung zurück auf einen Zustand vor dem Beschluss „Seitenpuffer“ des BGH aus dem Jahre 1991.³² Erwartungsgemäß werden auch Ansprüche auf ein auf einem Datenträger gespeichertes Programm allein nach Art. 5(1a) ausgeschlossen.

Zu diesem engen Technikverständnis gehört auch, dass nach Erwägungsgrund 13a eine computerimplementierte Geschäftsmethode grundsätzlich nicht-technisch sein soll. Am obigen Beispiel des Falls „Suche fehlerhafter Zeichenketten“ sollte gezeigt werden, dass es sehr wohl einen Unterschied macht, ob ein Verfahren mit Papier und Bleistift oder mittels eines Computers ausgeführt wird. Obwohl man insoweit auch einen technischen Beitrag zum Stand der Technik annehmen könnte, erscheint eine solche Interpretation nach der Intention der Entschließung als eher unwahrscheinlich.

Wenn in Art. 4 für das Kriterium der erfinderischen Tätigkeit ein „technischer Beitrag“ gefordert wird, so entspricht das zunächst dem hier vertretenen Standpunkt. Allerdings wird in Erwägungsgrund 11 für die erfinderische Tätigkeit „zusätzlich ein neuer technischer Beitrag zum Stand der Technik“ gefordert. Hier werden die Merkmale der Erfindungshöhe und der Neuheit vermischt, was sehr an die frühere „Kertheorie“ des BGH erinnert. Auch in Bezug auf die Erfindungshöhe wird in Art. 4 (3) eine Abkehr von einer Gesamtbetrachtung hin zu einer Konzentration auf die technischen Merkmale vollzogen.

Art. 6 eröffnet scheinbar ein verfassungsrechtlich problematisches Primat des Urheberrechtsschutzes gegenüber dem Patentschutz, indem die in Art. 5, 6 Software-RL (§§ 69d, e UrhG) festgelegten Schranken bei kollidierendem Patentschutz vorrangig sein sollen. Allerdings kann man die Voraussetzung des „rechtmäßigen Erwerbers/Nutzers“ in § 69d UrhG auch so verstehen, dass dieser das Programm ohne Verstoß gegen das Patentrecht erworben hat, sodass die Konfliktlage dann weitestgehend entschärft ist. Wie die Ausnahme vom Schutzzumfang „für einen bedeutsamen Zweck“ in Art. 6a verfassungskonform zu handhaben ist, erscheint ebenfalls unklar.

Was bleibt an Positivem zu berichten? Zu begrüßen ist die Klarstellung in Erwägungsgrund 13c, wonach ein Algorithmus grundsätzlich nichttechnischer Natur ist und nur bei Anwendung zur Lösung eines technischen Problems patentierbar ist und nur in diesem Kontext geschützt wird. Allerdings hängt die Reichweite dieser Freistellung entscheidend vom Begriffsverständnis ab.

Zu begrüßen ist ebenfalls der Hinweis darauf, dass die Schutzvoraussetzungen der Neuheit und erfinderischen Tätigkeit genau zu beachten sind.³³ Sollte es danach überhaupt noch computerimplementierte Erfindungen geben können, so ist die Forderung nach der Veröffentlichung einer „gut dokumentierte(n) Referenzimplementierung eines solchen Programms als Teil der Beschreibung ohne einschränkende Lizenzbedingungen“ nach Art. 5 (1d) zu fordern.

Insgesamt kann man konstatieren, dass die Entschließung wenig Spielraum für computerimplementierte Erfindungen lässt und die Rechtsentwicklung um mindestens zehn Jahre zurückwerfen wird. Es wird wieder das Problem der Abgrenzung

³² BGH CR 1991, 658.

³³ Vgl. Erw. grd. 18a.

technischer von nichttechnischen Programmen auftauchen, da die Richtlinie das Bedürfnis nach Patentierung in diesem Bereich nicht einfach vom Tisch wischen kann. Entgegen der ausdrücklichen Intention der Entschließung in Erwägungsgrund 5a wäre damit wieder große Rechtsunsicherheit geschaffen, die eine der größten Hindernisse für innovative Tätigkeit darstellt. Die Auswirkungen auf den Schutzzweck des Patentrechts sind unabsehbar.

III. Patentschutz und Open-Source-Software: Konfliktlage

Neben der Problematik der Geschäftsmethoden hat sich die Diskussion zur Frage Patentschutz und Software vor allem auf den Open-Source-Bereich zugespitzt. Geht man davon aus, dass es sich bei Open Source um ein alternatives Entwicklungs- und Vertriebsmodell handelt,³⁴ dass das Ausschließlichkeitsprinzip des Immaterialgüterrechts grundsätzlich in Frage stellt, dann kann nur eine völlige Ablehnung eines Patentschutzes für diesen Bereich die Folge sein.³⁵ Anlass genug, einmal näher zu untersuchen, welche Auswirkungen ein Patentschutz für Software auf Entwickler und Nutzer in diesem Bereich haben kann.³⁶

1. Grundsätzliche Bedenken gegen den Patentschutz

Nachdem oben die Vorteile des Patentschutzes herausgestellt wurden, sollen auch die immer wieder vorgebrachten Befürchtungen, insbesondere im Hinblick auf Open-Source-Entwickler, nicht verschwiegen werden. Mit der Erfassung von Algorithmen im konkreten Sinne ist die Befürchtung verbunden, dass die Software-Entwicklungstätigkeit durch ein „dichtes Gestrüpp von patentierten Algorithmen“³⁷ erschwert werde, die die Funktionalität von Programmen in vielen Bereichen abdecken und damit das Entwickeln von Programmen in diesem Bereich lizenzpflichtig machen könnten. Diese Gefahr besteht aber nur bei zu breiter Patentierung, wie sie vor allem in den USA vorgekommen ist. Bei sorgfältiger Patentprüfung und gut dokumentiertem Stand der Technik dürfte die Gefahr gering sein, dass die Programmierung ernsthaft behindert wird, auch wenn Programmierer oft ins Feld führen, dass ihre Tätigkeit sich von herkömmlicher technischer Entwicklungstätigkeit grundsätzlich unterscheide.

Ein weiteres Problem besteht darin, dass eine Recherche auf Grund der vielfältigen Verletzungsformen schwierig ist und das daraus entstehende Patentverletzungsrisiko vor allem für kleinere und mittlere Unternehmen schwer zu tragen sein kann.

³⁴ Lutterbeck/Gehring/Horns, Sicherheit in der Informationstechnologie und Patentschutz für Software-Produkte – Ein Widerspruch?, Kurzgutachten, Berlin 2000, <http://www.sicherheit-im-internet.de/download/Kurzgutachten-Software-patente.pdf>, S. 61 ff.

³⁵ Vgl. Lutterbeck, Jefferson revisited/ revised: Demokratische Technologie und Softwarepatente sind ein Widerspruch!, <http://ig.cs.tu-berlin.de/bl/088/Lutterbeck-FIFFJefferson-2003.pdf>.

³⁶ Vgl. auch Gehring/Lutterbeck, Software-Patente im Spiegel von Softwareentwicklung und Open Source Software, in: Taeger/Wiebe (Hrsg.), Recht-Wirtschaft-Informatik, Festschrift für Wolfgang Kilian, Baden-Baden 2004, S. 301 ff.; Wiebe, in: Spindler (Hrsg.), Rechtsfragen bei Open Source, Köln 2004, Kap. F Rn. 43 ff.

³⁷ Metzger, Risiko Softwarepatente?, http://www.ifrOSS.de/ifrOSS_html/art11.pdf, S.2. Vgl. auch Lutterbeck, in: Taeger/Wiebe (Hrsg.), S. 301, 307 ff., mit Hinweis auf die „Granularität“ von Software und die zunehmende Erfassung von Basistechnologie.

Bei den Recherche-Problemen handelt es sich jedoch um typische Anfangsprobleme bei neuen Technologien. Zudem ist es Open-Source-Entwicklern als Abwehrstrategie möglich, durch Einrichtung von Dokumentationsstellen, in denen Softwareentwickler ihre Arbeitsergebnisse mit Zeitstempel registrieren und der Öffentlichkeit zugänglich machen können, den Stand der Technik zu bereichern und damit der Erteilung von Patenten auf solche Software entgegenzuwirken.³⁸

Bei Open Source besteht auf Grund der Verfügbarkeit des Quellcodes der Nachteil, dass die Feststellung einer Patentverletzung einfacher wird und damit die Rechtsverfolgung durch „proprietär“ ausgerichtete Unternehmen erleichtert wird, die selbst ihre Programme häufig nur im Objektformat verbreiten, dessen Rückentwicklung durch das Urheberrecht verhindert wird. Andererseits ist wegen der funktionalen Ausrichtung des Patentschutzes häufig die Verfügbarkeit des Quellformats noch nicht ausreichend, um eine Patentverletzung festzustellen.

Eine weiterer Nachteil gegenüber „proprietären“ Anbietern ist die Schwierigkeit einer aktiven Patentrechtspolitik („konzeptionelle Asymmetrie“ des OSS-Modells).³⁹ Patentrecherche und Patentstreitigkeiten sind kostenintensiv und von den meist kleinen oder mittelständischen Entwicklern und Distributoren finanziell schwerer durchzustehen. „Kreuzlizenzen“ sind eine häufige Form der Patentverwertung: mit eigenen Lizenzen als „Währung“ erwirbt man Zugang zu Technologien der Mitbewerber. Dieser Weg ist Open-Source-Entwicklern weitgehend verwehrt.

Schließlich wird auch ins Feld geführt, dass der Patentschutz zu einer Blockade von Standards führe, indem kleinere, nicht zu einem Patentpool gehörende Unternehmen am Zugang gehindert werden. Praktisch hat sich das aber bisher kaum als Problem erwiesen.⁴⁰

2. Verletzung bestehender Patente durch Open-Source-Entwickler

Auf dem Hintergrund dieser allgemeinen Bedenken stellt sich insbesondere die Frage, inwieweit die Softwareentwicklung selbst durch einen Patentschutz behindert wird. Dazu sind die verschiedenen Konfliktsituationen näher zu beleuchten. Im Vordergrund steht zunächst die Situation, dass ein Open-Source-Entwickler Teile einer patentrechtlich geschützten softwareimplementierten Erfindung nutzt und auch weiterverbreitet. Hier stellt sich die Frage, inwieweit der Schutzbereich des „proprietären“ Programms berührt ist.

a) Patentverletzung §§ 9, 10 PatG

Zumeist ist der softwarebezogene Patentanspruch als Verfahrenserfindung gefasst, sodass sich eine Patentverletzung auf die Anwendung des Verfahrens sowie das Anbieten zur Anwendung beschränkt. Wegen der Ausrichtung auf die Funktionalität geht es bei der Patentverletzung nicht um die Ähnlichkeit der Programmbeefehle, sondern um eine Ähnlichkeit der „dynamischen“ Funktionsabläufe.⁴¹ Da man sich nicht allein auf die Quellprogramme stützen kann, erschwert dies die Recherche

³⁸ Vgl. Lutterbeck/Gehring/Horns, S. 11 f.; Jaeger/ Metzger, Open Source Software, München 2002, S. 112.

³⁹ Horns, www.jurpc.de/aufsatz/20000223.htm, Abs. 55.

⁴⁰ Vgl. Haase, S. 178.

⁴¹ Vgl. auch Heide, CR 2003, 165, 167.

des Stands der Technik.⁴² Gerade auf Grund der Kennzeichnung von Ansprüchen durch funktionale Merkmale kann der Schutzzumfang besonders durch eine übermäßig breite Fassung der Ansprüche erweitert werden.

Neuerdings gewährt die Rechtsprechung auch Erzeugnispatente auf das Computerprogramm selbst.⁴³ Der Anspruch ist dann auf ein digitales Speichermedium gerichtet, auf dem das Programm gespeichert ist. Dies soll jedenfalls insoweit patentierbar sein, als es als Mittel zur Ausführung eines Verfahrens dient und dieses Verfahren technischen Charakter aufweist.⁴⁴ Dann stellt bereits die Herstellung, das Anbieten und Inverkehrbringen, auch über Internet, eine Rechtsverletzung dar, was eine erhebliche Ausweitung des Schutzes bedeutet und auch die Weiterverbreitung von Open-Source-Software erschwert.⁴⁵

Allerdings kommt es auf die Fassung der Ansprüche an, ob das Anbieten der Software sich nur auf einen Teil der Erfindung bezieht und damit in den Bereich der mittelbaren Patentverletzung gem. §10 PatG fällt.⁴⁶ Auch bei einer Verfahrenserfindung kann das Computerprogramm ein Teilsystem darstellen, das ein wesentliches Element des geschützten Verfahrens konstituiert.⁴⁷ Für eine Rechtsverletzung bedarf es hier besonderer subjektiver Voraussetzungen.

Aus einer Patentverletzung können sich Unterlassungs- und Schadensersatzansprüche ergeben. Auch wenn eine Haftung auf Schadensersatz Verschulden voraussetzt und von einem einfachen Entwickler keine breite Patentrecherche verlangt werden kann,⁴⁸ ist doch zu beachten dass die einschlägigen Patente in Programmiererkreisen zumindest soweit bekannt sein können, dass Anlass zu weitergehender Recherche besteht.

b) Ausnahmetatbestände § 11 PatG

Selbst wenn die Entwicklung und Verbreitung von Open-Source-Software ein bestehendes Patent verletzen sollte, kann es sein, dass eine der Ausnahmen vom Schutzzumfang greift. Nach §11 Nr. 1 PatG gehören dazu Handlungen im privaten Bereich zu nichtgewerblichen Zwecken. Hierunter fällt also die private Nutzung von Open-Source-Software. Beim Austausch oder Vertrieb von Software über das Internet wird aber wohl die Grenze des privaten Bereichs überschritten.⁴⁹ Bei einer Verfahrenserfindung bleibt zu beachten, dass ein Anbieten zu erlaubten Zwecken nach §9 Abs. 1 Nr. 2 PatG möglich bleibt. Dies wäre eigentlich bei einer Weitergabe an einen Nutzer zu dessen privatem Gebrauch gegeben. Allerdings werden dem

⁴² Vgl. Horns, www.jurpc.de/aufsatz/20000223.htm, Abs. 53.

⁴³ EPA, CR 2000, 91, 93 – Computerprogrammprodukt/IBM; BGH, GRUR Int. 2002, 323 – Suche fehlerhafter Zeichenketten. S.o. Fußn. 20 f.

⁴⁴ BGH, Beschluß v. 17.10.2001 – X ZB 16/00, GRUR Int. 2002, 323 – Suche fehlerhafter Zeichenketten. Kritisch dazu Vorschlag für eine Richtlinie des Europäischen Parlaments und des Rates über die Patentierbarkeit computerimplementierter Erfindungen, KOM(2002) 92 endg. v. 20.2.2002, Begründung, S. 16.

⁴⁵ Vgl. Esslinger/Betten, CR 2000, 18, 20.

⁴⁶ Vgl. Heide, CR 2003, 165, 166.

⁴⁷ Vgl. Lutterbeck/Gehring/Horns, S. 98 ff., die als Beispiel das separate Anbieten eines Speicherbausteins für eine Kraftfahrzeugbremsensteuerung nennen.

⁴⁸ So auch Metzger, http://www.ifrOSS.de/ifrOSS_html/art11.pdf, S. 2.

⁴⁹ Vgl. auch Esslinger/Betten, CR 2000, 18, 21.

Nutzer keine Weitergabebeschränkungen auferlegt, sodass eine Eingrenzung auf den privaten Gebrauch nicht gewährleistet werden kann, und damit wohl doch der Verletzungstatbestand erfüllt wäre.⁵⁰

Freigestellt werden durch §11 Nr. 2 PatG weiterhin Handlungen zu Versuchszwecken, die sich auf den Gegenstand der patentierten Erfindung beziehen. Obwohl die Rechtsprechung die Grenzen dieses Ausnahmetatbestands relativ eng zieht, dürfte die Open-Source-Entwicklung darunter fallen, wenn man das Prinzip darin sieht, dass ständig Verbesserungen an der geschützten Software vorgenommen werden. Diese Schranke dient der Forschung und der technischen Innovation. Allerdings darf eine resultierende Entwicklung nicht in den Schutzbereich bestehender Patente eingreifen. Zu beachten ist dabei auch, dass bei parallel bestehendem Urheberrecht ein „reverse engineering“ nur in den engen Grenzen von §69e UrhG zulässig ist.⁵¹

3. Schutz gegen Patentanmeldungen durch „proprietary“ ausgerichtete Softwareunternehmen

Möglich ist, dass „proprietary“ ausgerichtete Unternehmen Patente anmelden, deren wesentliche Bestandteile aus Open-Source-Software bestehen. Eine zentrale Abwehrstrategie der Open-Source-Entwickler besteht hier darin, dass die Software möglichst schnell veröffentlicht wird, um somit die Neuheit einer Erfindung nach §3 PatG auszuschließen. Dabei reicht eine Veröffentlichung irgendwo auf der Welt bereits aus, sodass auch eine Zugänglichmachung des Quellcodes über Internet ausreicht, wenn es nicht um private E-Mails geht. Um Beweisschwierigkeiten zu vermeiden, erscheint die Einrichtung einer Dokumentationsstelle sinnvoll.⁵²

Einen Schutz gegen einen Patentinhaber bietet auch die Berufung auf das Vorbenutzungsrecht des §12 PatG. Das setzt voraus, dass der Open-Source-Entwickler die Erfindung erkannt hat und diese vor der Patentanmeldung bereits benutzt oder die zur baldigen Anwendung notwendigen Maßnahmen getroffen hat. Hat der Open-Source-Entwickler die Kenntnis vom späteren Schutzrechtsinhaber erlangt, entsteht ein Vorbenutzungsrecht nicht, wenn er sich die Kenntnis widerrechtlich verschafft hat.⁵³ Ein Vorbenutzungsrecht verleiht einen fortbestehenden Anspruch auf Benutzung der Erfindung im Inland. Eine entscheidende Einschränkung ergibt sich jedoch aus der Beschränkung auf den eigenen Betrieb. Eine solche betriebsbezogene Verwertung wird für das Open-Source-Prinzip nur dann anzunehmen sein, wenn die Entwicklung innerhalb fest organisierter Communities erfolgt und neu entwickelte Versionen vor ihrer Verbreitung zunächst durch den „Kopf“ der Organisation genehmigt werden müssen.

Relevant erscheint auch der umgekehrte Fall, nämlich dass der „proprietary“ Entwickler die patentierte Lehre einem Open-Source-Programm entnommen und dann selbst angemeldet hat. Nach §6 PatG steht das Recht auf das Patent dem Erfinder zu. Hat der Anmelder den wesentlichen Inhalt der Erfindung Beschreibun-

⁵⁰ Vgl. auch Jaeger/Metzger, S. 119.

⁵¹ Vgl. auch Jaeger/Metzger, S. 117.

⁵² Vgl. Jaeger/Metzger, S. 114 ff.

⁵³ Vgl. BGH GRUR 1964, 675.

gen oder Unterlagen des Erfinders ohne dessen Einwilligung entnommen, kann das Patent auf Einspruch hin widerrufen werden (§21 Abs. 1 Nr. 3 PatG).⁵⁴ Eine bloße Mitteilung oder Überlassung zum Nachbau beinhaltet noch keine Einwilligung zur Patentanmeldung.⁵⁵ Auch hier hat ein Open-Source-Entwickler also ein Abwehrmittel in der Hand, vorausgesetzt, die Open-Source-Software und die später angemeldete Erfindung stimmen in ihrem wesentlichen Inhalt überein.

4. Schutz gegen Patentanmeldungen durch Open-Source-Entwickler

Schließlich droht der Open-Source-Gemeinde auch „Gefahr“ aus den eigenen Reihen, wie der Fall „RTLinux“ gezeigt hat.⁵⁶ Kann ein Patentanmelder einer Open-Source-Software mittels der Lizenz verpflichtet werden, die Software frei zu lizenzieren? Die Reichweite etwa der GNU-Lizenz ist davon abhängig, inwieweit Weiterentwicklungen, die möglicherweise patentfähig sind, im urheberrechtlichen Schutzbereich liegen.⁵⁷ Nur wenn urheberrechtlich relevante Nutzungshandlungen bei der Entwicklung des Derivats erfolgen, ergibt sich aus der GNU-GPL eine Verpflichtung, das Derivat auch patentrechtlich unter die Bedingungen der Lizenz zu stellen.⁵⁸ Die „Open RTLinux Patent Licence“⁵⁹ erlaubt die lizenzgebührenfreie Nutzung des Patents entsprechend auch nur, wenn GPL-Software eingesetzt wird. In anderen Fällen ist eine „proprietäre“ Lizenz einzuholen.⁶⁰

IV. Lösungsvorschläge

Verbleibt somit ein Konfliktpotential, so werden verschiedene Wege zur Lösung diskutiert.

1. Schutzzweckbezogene Auslegung des Patentgesetzes

Ein wichtiger Eckpfeiler dafür, dass das Patentrecht wirklich die ihm zugeordnete Wirkung der Förderung der Innovation erfüllen kann, ist die korrekte Anwendung der gesetzlichen Voraussetzungen. Dazu gehört, dass hohe Anforderungen an das Vorliegen der Erfindungshöhe und Neuheit zu stellen sind.⁶¹ Weiterhin ist zu beachten, dass auch im Patentrecht der Grundsatz der Schutzfreiheit allgemeiner Lehren und Methoden gilt (§1 Abs. 2 Nr. 1 PatG). Richtig angewendet, müssen danach auch Algorithmen einer bestimmten Abstraktionsstufe frei bleiben.⁶²

Möglich ist auch eine Begrenzung des Schutzzumfangs softwarebezogener Patente. Zwar entspricht die Einbeziehung äquivalenter Lösungen ständiger Rechtspre-

⁵⁴ Vgl. auch Jaeger/Metzger, S. 116.

⁵⁵ Vgl. Benkard, PatG, §9 Rn. 9.

⁵⁶ Vgl. Jaeger/Metzger, S. 121.

⁵⁷ Vgl. dazu Wuermeling/Deike, CR 2003, 87, 88 ff.; Jaeger/Metzger, S. 123 ff.; Jaeger, in: Wiebe/Leupold (Hrsg.), Recht der elektronischen Datenbanken, Heidelberg 2002, III-B Rn. 79 ff.

⁵⁸ Jaeger/Metzger, S. 124, meinen darüber hinaus, dass §2b) GPL auch Patente erfasst, wenn zugunsten freier Entwicklergruppen bereits Patente bestehen, was aber kaum praktisch werden dürfte.

⁵⁹ Vgl. http://www.fsmlabs.com/products/rtnlinuxpro/rtnlinux_patent.html.

⁶⁰ So auch Jaeger/Metzger, S. 125 f., mit Hinweis auf die 2. Version der Open RTLinux Patentlizenz.

⁶¹ Gerade die Neuheit der in den USA erteilten Patente wird immer wieder bezweifelt, vgl. etwa zum RTLinux-patent Jaeger/Metzger, S. 121 Fußn. 517.

⁶² Vgl. Wiebe, in: Spindler (Hrsg.), Rechtsfragen bei Open Source, Kap. F Rn. 40 ff.

chung des BGH und hat eine Grundlage auch im Auslegungsprotokoll zu Art. 69 EPÜ. Allerdings sollten die Gerichte bei der Bestimmung der technischen Gleichwirkung nicht zu großzügig vorgehen und die Besonderheiten der Informatik im Sinne handhabbarer Abgrenzungen berücksichtigen. Dadurch könnten in Kombination mit den Ausnahmen von §11 Nr. 1, 2 PatG Freiräume geschaffen werden.

2. Quelltext und Waffengleichheit

Vorgeschlagen ist weiterhin, den Vertrieb von Software im Quelltext vom Schutzzumfang des Patentrechts auszunehmen.⁶³ Damit wäre ein Teil des Open-Source-Entwicklungsmodells von der Behinderung durch den Patentschutz befreit.

Ein interessanter Vorschlag ist auch, „Waffengleichheit“ durch eine Verpflichtung „proprietärer“ Software-Patentanmelder zur Hinterlegung eines Programmlistings herzustellen.⁶⁴ Damit hätten die Open-Source-Entwickler noch besseren Zugang zum patentierten technischen Wissen. Andererseits besteht natürlich ein gravierendes Geheimhaltungsinteresse, soweit dies im Hinblick auf den Offenbarungsgrundsatz möglich ist.⁶⁵ Ein interessanter Weg kann jedoch in der in Art. 5 (1d) der Entschließung des Europäischen Parlaments vorgeschlagenen „Referenzimplementierung“ liegen.⁶⁶

3. Anwendung der Zwangslizenzregelung (§ 24 PatG)

Die Möglichkeit der Erteilung einer Zwangslizenz ist eher ein seltener Ausnahmefall und das dafür notwendige öffentliche Interesse wurde bisher nur sehr restriktiv angenommen. Es ist eine auf den Einzelfall zugeschnittene Ausnahmeregelung, muss durch Klage (§ 81 Abs. 1 PatG) durchgesetzt werden und ist insoweit praktisch zur Absicherung des Open-Source-Modells nicht geeignet.

4. Patent Pool – die kollektive Lösung

In Anlehnung an die Patentrechtspolitik größerer Unternehmen wird für Open-Source-Software die Bildung von Patentpools vorgeschlagen, die der Abwehr von Patentverletzungen und der Kreuzlizenzierung mit einer breiteren Basis dienen könnten.⁶⁷ Voraussetzung eines solchen Pools ist aber der Erwerb eigener Patente. Dies ist allerdings nach der derzeitigen Gesetzeslage im Hinblick auf die Funktionsweise des Open-Source-Modells weitgehend ausgeschlossen. Zumindest eine gewisse Erleichterung bringen Modelle des „Dual Licensing“, wie von RTLinux praktiziert, sowie der Eclipse-Lizenz von IBM, die in einem Projekt entwickelte Software von Patentansprüchen gegen die entwickelte Software freistellt.⁶⁸

⁶³ Vgl. Lutterbeck/Gehring/Horns, S. 132: „Herstellen, Anbieten, In-Verkehr-bringen, Besitzen oder Einführen von Software im Quellcode“.

⁶⁴ Vgl. Haase, S. 175.

⁶⁵ Vgl. Stobbs, Software Patents, New York 1995, S. 170, wonach der Source Code für Zwecke der Offenbarung einerseits zu detailliert ist, andererseits die Erfindung nicht genau beschreibt.

⁶⁶ S. o. unter II.2.d).

⁶⁷ Vgl. Gehring, Berliner Ansatz zu „Open Software Patents“, <http://ig.cs.tu-berlin.de/ap/rg/2000-05/Gehring-OpenSoftwarePatents-052000.pdf>.

⁶⁸ Gehring/Lutterbeck, in: Taeger/Wiebe (Hrsg.), S. 301, 316.

V. Fazit

Die genauere Betrachtung hat ergeben, dass es durchaus Konfliktpotential zwischen Patentschutz und Softwareentwicklung gibt. Andererseits besteht eine entsprechende Konfliktlage auch für andere patentierte technische Entwicklungen. Insbesondere die nähere Untersuchung der Open-Source-Software hat aufgezeigt, dass es durchaus bereits nach der bestehenden Rechtslage Möglichkeiten gibt, ohne Beeinträchtigung durch bestehenden Patentschutz Softwareentwicklung zu betreiben. Weitere Probleme sind durch mangelnde Erfahrung und einen noch nicht ausreichend dokumentierten Stand der Technik in diesem Bereich bedingt. Diese werden sich mit der Zeit abschwächen, wozu durch Dokumentation und Einrichtung entsprechender Stellen beigetragen werden kann.⁶⁹ Schließlich ist das Dual Licensing Indiz dafür, dass ein gewisser Wettbewerb möglich erscheint.⁷⁰

Die größte Gefahr scheint von einer zu breiten Patentierung unter Einschluss von Algorithmen auszugehen. Gerade insoweit sind die bestehenden Grundsätze der Patentierung besonders zu beachten. Insbesondere ist die Rechtsprechung in Zukunft gefordert, noch stärker die Grenzlinie zwischen schutzfreien Algorithmen und geschützten innovativen Programmfunktionen herauszuarbeiten. Auch sind die hohen Anforderungen an Neuheit und erfinderische Tätigkeit strikt zu beachten. Unter diesen Voraussetzungen steht zu erwarten, dass die Patentierung computerimplementierter Erfindungen in gleichem Masse eine innovationsfördernde Wirkung haben können wie herkömmliche technische Erfindungen. Insoweit gelten die eingangs angeführten Vorteile der Patentierung. Das gilt gerade auch für den Markteinstieg von KMU, für den verschiedene Studien die positiven Effekte der Patenterteilung hervorgehoben haben.⁷¹ Es bleibt zu hoffen, dass wir durch die Entschließung des Europäischen Parlaments zur Harmonisierung im Bereich computerimplementierter Erfindungen nicht in den früheren Zustand der Rechtsunsicherheit zurück geworfen werden.

⁶⁹ Vgl. auch die einerseits eher zuversichtliche Schlußfolgerung von Jaeger/Metzger, S. 127 f., andererseits die skeptische von Gehring/Lutterbeck, in Taeger/Wiebe (Hrsg.), S. 301, 321

⁷⁰ Vgl. auch Haase, S. 175.

⁷¹ Vgl. Haase, S. 138 ff., m.w.Nachw.