

Mit der Informationstechnik ordnen wir die informationelle Umwelt des Menschen völlig neu, neben das Buch tritt die Datenbank, das Expertensystem, neben die klassische Datenübertragung von Mensch zu Mensch tritt die technische Kommunikation, das Wachstum an Daten und Information wird durch die Technik rapide beschleunigt. Es ist eine wichtige Aufgabe, unsere informationelle Umwelt angemessen zu ordnen, so daß der Mensch sich in ihr zurechtfindet und in ihr zufrieden leben kann. Ähnlich wie wir im Bereich der materiellen Umwelt erhebliche Umweltschäden durch unreflektierten Umgang mit der Natur bewirkt haben, so wird ein unkritisches und vor allen Dingen unwissendes Umgehen mit der Informationstechnik gravierende Konsequenzen für unsere infor-

mationelle Umwelt haben. Dem gilt es dadurch entgegenzuwirken, daß breite Schichten über diese Technik aufgeklärt und in den Stand versetzt werden, mit ihr bewußt und kritisch umzugehen.

Literatur

Haefner, K.: Der große Bruder — Chancen und Gefahren für eine informierte Gesellschaft. Düsseldorf 1980

Haefner, K.: Die neue Bildungskrise — Herausforderung der Informationstechnik an Bildung und Ausbildung. Basel 1982 und Reinbek 1985

Haefner, K.: Mensch und Computer im Jahre 2000 — Ökonomie und Politik für eine human computerisierte Gesellschaft. Basel 1984

Die Werkhöhe bei Software

Ing. Werner Schmidt und
Rechtsanwalt Dr. Günter Knorr

I. Die Rüge des Bundesgerichtshofs

Der vielzitierte Musterprozeß über den Urheberrechtsschutz für Software ist bekanntlich vom Bundesgerichtshof in seiner Entscheidung vom 9. 5. 1985¹ nur zwischenbeschieden worden: er wurde an das Berufungsgericht zur anderweitigen Verhandlung und Entscheidung zurückgewiesen. Bemerkenswert ist die Begründung: nachdem der BGH die Urheberrechtsschutzfähigkeit von Computerprogrammen grundsätzlich bejaht hat, wendet er sich der wesentlich bedeutsameren Frage zu, wie diese Erkenntnis in die fallentscheidende Praxis umgesetzt werden könne. Dabei rügt der Bundesgerichtshof die mangelnde Tatsachenerforschung durch das OLG Karlsruhe. Dieses hatte sich aus der Affaire gezogen, indem es die persönliche geistige Leistung des Inkassoprogramms, um das die Parteien stritten, recht knapp behauptete: „Die Verwendung einer Systemhülle, wie es in dem von der Klägerin vorgelegten Gutachten Mertens/Bodendorf über die Individualität des klägerischen Computerprogramms treffend heißt, läßt die persönliche geistige Leistung bei der Gestaltung des Programms in der Sammlung und Zuordnung des verarbeiteten Stoffs erkennen. Die Kombination der außergerichtlichen Mahnung und des gerichtlichen Mahnverfahrens mit verschiedenen Kontrollmechanismen gibt der Gestaltung des X-Inkassoprogramms insgesamt eine schöpferische Prägung.“² Zum Beleg dieser Behauptung werden praktische Vorzüge des Programms aneinandergereiht, die „jedenfalls“ in ihrer Kombination ein prägendes Gewebe individueller Leistung ergebe. Zu pauschal und nicht wirklich begründet, erkennt mit Recht der BGH: „Der Senat vermag mangels entsprechender Feststellungen nicht zu überprüfen, ob eine deutlich über dem Durchschnitt liegende schöpferische

Leistung vorliegt.“ Schließlich muß, das hatte der BGH in seinen allgemeinen Ausführungen zuvor als seine ständige Rechtsprechung betont, das allgemeine Durchschnittskönnen eines Programmierers in Auswahl, Sammlung, Anordnung und Einteilung der Informationen und Anweisungen deutlich überragt werden.

II. Der Hintergrund der Rüge

Allerdings: die Rüge ist leichter als ihre Umsetzung in praktische Entscheidungen. Wie aber Grundsätze des Urheberrechts für die Werkart Software in die Praxis umgesetzt werden können, das entscheidet darüber, ob das herkömmliche Rechtsgebiet dem neuen immateriellen Gut Software ausreichende Protektion gewähren kann.

Ein kurzer Rückblick auf geschichtliche Wurzeln des deutschen Urheberrechts kann zur Gewinnung der Diagnose nichts schaden: das deutsche Urheberrecht ging ursprünglich vom Problem des Büchernachdrucks aus³. Die Lehre vom geistigen Eigentum, die erstmalig im 18. Jahrhundert auftauchte⁴, bedeutete den Beginn der Lehre, die das Schutzobjekt „Werk“ vom konkreten Werkstück unterscheidet. Derartige Entwicklungen, die uns heute als bedeutsame Errungenschaften

Anmerkungen:

¹ BB 1985, S. 1747

² GRUR 1983, S. 306/7

³ ausführlich dargestellt bei *Vogel*, Deutsche Urheber- und Verlagsrechtsgeschichte zwischen 1450 und 1850, Frankfurt 1978

⁴ *Gieseke*, Die geschichtliche Entwicklung des deutschen Urheberrechts, Göttingen 1957, S. 72 ff.

des Urheberrechts erscheinen⁵, führten trotz jahrzehntelanger Systematisierungs- und Durchdringungsversuche zu größeren Definitionsproblemen und Schwierigkeiten, wenn sie in die Praxis umgesetzt werden sollten. Die idealistische und immer wieder für selbstverständlich gehaltene Begründung des „geistigen Eigentums“ erforderte einen des Schutzes würdigen Gegenstand. Es wurden und werden also für die Scheidung von schutzwürdigen und nichtschutzwürdigen Werken Kriterien benötigt, die den Schutz auch im Einzelfall rechtfertigen können.

Heute ist auf der Grundlage der Lehre vom Immaterialgüterrecht anerkannt, daß es die „Individualität“ eines Werkes ist, die seine Schutzwürdigkeit ausmacht⁶, das Besondere⁷, wobei es das Ergebnis einer persönlichen, also menschlichen geistigen Schöpfung sein muß und auch von der Persönlichkeit des Urhebers geprägt sein muß⁸. Dabei ist im Auge zu behalten, daß das Werk als etwas Geistiges empfunden wird, das in dem Werkstück nur seinen Ausdruck findet, ohne mit ihm identisch zu sein. Die Individualität des Werkes spiegelt die Individualität seines Schöpfers. Diese Lehre ist vom Geniebegriff des 18. Jahrhunderts abgeleitet, wobei eine Analyse des Schaffens nicht stattfindet. Kant, der die Diskussion ausgangs des 18. Jahrhunderts maßgeblich beeinflusst hat, schreibt in seiner „Kritik der Urteilskraft“: „Man sieht daraus, daß Genie I.) ein Talent sei, dasjenige, wozu sich keine Regel geben läßt, hervorzubringen; nicht Geschicklichkeitsanlage zu dem, was nach irgendeiner Regel gelernt werden kann; folglich daß Originalität seine erste Eigenschaft sein müsse . . . — III.) daß es, wie es sein Produkt zustandebringe, selbst nicht beschreiben oder wissenschaftlich anzeigen könne, sondern daß es als Natur die Regel gebe; und daher der Urheber eines Produkts, welches er seinem Genie verdankt, selbst nicht weiß, wie sich in ihm die Ideen dazu herbeifinden, auch es nicht in seiner Gewalt hat, dergleichen nach Belieben oder planmäßig auszudenken . . .“⁹.

Daß eine weitere Erforschung des Schöpfungsvorgangs zwar wünschenswert, aber kaum möglich ist, zeigt die Lehre von „Form“ und „Inhalt“. Nach ihr sollte am Werk die Form geschützt, der Inhalt jedoch frei sein. Es stellte sich heraus, daß eine derartige Unterscheidung nicht wirklich durchzuführen ist, man mußte deshalb mit einem Kunstgriff die sogenannte „innere Form“ mit einbeziehen¹⁰. Die heutige Lehre scheidet deshalb die „individuellen Züge“ des Werkes gegenüber dem literarischen und künstlerischen Allgemeingut¹¹. Diese Unterscheidung hat zwar die theoretische und logische Überzeugungskraft für sich, bedeutet jedoch für die Praxis deshalb eine Überforderung, weil niemand in der Lage sein wird, das gesamte „Allgemeingut“ wirklich zu kennen. Wie uns jetzt wieder der BGH gelehrt hat, ist dies die Voraussetzung dafür, um im praktischen Fall wirklich trennen zu können.

Das Denken des deutschen Urheberrechts und seine Wurzeln liegen im individuellen, nämlich der Beziehung des einzelnen Autors zu seinem Werk; aus dieser Beziehung fließt, nach der monistischen Lehre die

ideelle und materielle Befugnis des Autors an seinem Werk; das Urheberrecht ist ein „Immaterialgüterrecht“ mit ausschließlicher, umfassender und einheitlicher Natur¹².

Unter den Vorgaben, daß der Algorithmus nicht schutzbar ist und auch nicht die bei der Erstellung des Programms herangezogenen mathematischen und technischen Lehren oder Regeln, sind deshalb die Leitlinien des BGHs für die Ermittlung einer zum Schutze ausreichenden Werkhöhe von Software bei dem oben geschilderten Stand der Lehre nicht überraschend. Nicht im Sinne einer Neuheitsprüfung, sondern einer Individualitätsprüfung muß von vorbekannten Programmen ausgegangen werden; ergibt das Programm auf dem Prüfstand daran gemessen schöpferische Eigenheiten, werden eben diese Eigenheiten dem Schaffen des Durchschnittsprogrammierers gegenübergestellt, um zu ermitteln, ob ein erheblicher Abstand zum rein Handwerksmäßigen, mechanisch-technischen Aneinanderreihen besteht. Unerheblich sind die Größe des Programms und der in ihm steckende Aufwand. Unerheblich ist auch, ob eine neue Aufgabe zur Lösung anstand und daß die Aufgabe verschieden gelöst werden konnte.

III. Der Programmvergleich in der Praxis

1. Die Scheidung des Allgemeinguts

Kann eine fachkundige Person, deren Wissen Parteien und Gericht in einer rechtlichen Streitigkeit für Sachvortrag und Entscheidung nützen können, nach heutigem Stand der Entwicklung den Sachverhalt so ermitteln, daß die tatsächlichen Feststellungen zur Grundlage der konkreten Entscheidung für oder gegen die Werkhöhe geschaffen werden? Zur vergleichenden Beurteilung angebotener Software sind eine Reihe von allgemein zugänglichen sogenannten Softwarekatalogen erhältlich¹³. In Verzeichnissen dieser Art sind Programme nach verschiedenen Sortierkriterien aufgeführt, etwa nach ihrem Einsatzgebiet wie „Rechnungswesen“, „Logistik“. Dazu kommt eine zumeist sehr knappe Funktionsbeschreibung. Aus dieser Beschreibung geht regelmäßig nur hervor, was das Programm verarbeitet, welche Art von Informationen ein- und ausgegeben werden. Eine Darstellung des „Wie“, nämlich der Art und Weise der Verarbeitung, findet sich nur in Ausnahmefällen.

⁵ Ulmer, Urheber- und Verlagsrecht, 3. A., Berlin/Heidelberg/New York 1980, S. 105

⁶ Ulmer, a. a. O., S. 123

⁷ Vinck bei Fromm/Nordemann: Urheberrecht, 5. A. Stuttgart 1983

⁸ Vinck a. a. O., § 2 Anm. 5

⁹ Kant, Kritik der Urteilskraft; herausgegeben von Wilhelm Weischedel, 2. A., Frankfurt a. M. 1977, S. 180 f.

¹⁰ vgl. die Darstellung bei Kohler, Urheberrecht an Schriftwerken und Verlagsrecht, Stuttgart 1907, S. 128 ff.

¹¹ Ulmer a. a. O., S. 122

¹² von Gamm, Urheberrechtsgesetz, München 1968, RZ 1 zu § 11

¹³ z. B. ISIS Software Report, München

Immerhin: anhand derartiger Kurzbeschreibungen kann ein Interessierter feststellen, ob Lösungen für eine bestimmte Problemstellung der Informationsverarbeitung bereits existieren. Soweit Kurzbeschreibungen nach den Vorgaben erstellt sind, die durch die Bestrebungen der Gütegemeinschaft Software e. V. in die DIN 66 285 unter dem Stichwort „Produktbeschreibung“ eingeflossen sind, geht aus ihnen auch mit größerer Deutlichkeit hervor, welche Funktionen ein Programm enthält. Damit ist eine Abgrenzung der einzelnen angebotenen Programme nach Funktions- und Aufgabenbereich gegeneinander möglich.

Das „Allgemeingut“ im urheberrechtlichen Sinne für den Bereich der Software ist jedoch mit diesen Beschreibungen nicht zu ermitteln. Die Ermittlung des Allgemeinguts ist — streng durchgeführt — wohl für fast alle Werkarten ein eher theoretisches Postulat, an dem sich Gerichte mit Sachverständigenhilfe gelegentlich versuchen müssen. Anders als bei einem Patent erlangt man den Schutz des Urheberrechts ja formlos — man dächte sich eine Behörde, die bei Urheberrechtsanmeldungen eine Individualitätsprüfung durchführt. Da es derlei nicht gibt, ist nur aus der Erfahrung im Umgang mit Programmen ein Überblick zu gewinnen, wie durchschnittliche Programmlösungen aussehen. Ob durch bevorzugt systematische Erforschung einer Programmoberfläche bei Praktikern verlässliches Know-how schon vorhanden ist, kann schwer beantwortet werden. Am ehesten wird sich ein breites Spektrum zum Vergleich bei Fachleuten ansammeln, denen systematische Programmprüfung Beruf ist. Derzeit entwickelt sich das Berufsbild des Softwareprüfers, das möglicherweise künftig wie das des Wirtschaftsprüfers in der Öffentlichkeit bekannt und akzeptiert sein wird. Diese Entwicklung läßt hoffen, daß das vergleichende Wissen von mehr und mehr Fachleuten mit größerer Sicherheit Allgemeingut der Programmierung von Leistungen individueller Prägung trennen kann.

2. Beurteilung der Individualität der Software

Die Individualität von Software hat enge Grenzen, die einerseits durch die Basisfunktion des umgebenden Computersystems (Zentralprozessor, Mikroprogramm, Betriebssystem) und andererseits durch die anwendungsspezifischen Anforderungen gegeben sind. Ein Programm zur Erstellung einer Rechnung („Fakturierung“) kann in seiner Kategorie nicht mehr leisten, als die einzelnen Rechnungsposten addieren, Zu- oder Abschläge verrechnen und den Endbetrag allenfalls für eine automatische Weiterverarbeitung in einem Buchhaltungsprogramm bereitstellen. Ein Programm zur Speicherung und Verwaltung großer Datenmengen, ein sogenanntes Datenbanksystem, verarbeitet alle Anweisungen des Datenbankanwenders nach einem dem Programm fest vorgegebenen Schema. Die verschiedenen Modelle zur logischen und physikalischen Anordnung großer Datenmengen in einem Computersystem sind in ihren Grundstrukturen der Praxis entlehnt. So sind beispielsweise alle Posten einer Buchhaltung streng hierarchisch gegliedert. Das Ord-

nungsschema wird vom Kontenplan bestimmt, der die Zugehörigkeit der Buchungsposten zu Einzelkonten, Sammelkonten oder Positionen in der Bilanz ausweist.

Individuelle Denkansätze zur Speicherung oder Verarbeitung von Daten durch die Programme stoßen naturgemäß an diese beiden Grenzen: entweder laufen sie Gefahr, nicht praxisgerecht zu sein, oder sie sind technisch nicht oder nicht in wirtschaftlich sinnvollen Rahmenbedingungen realisierbar, etwa weil die verfügbaren Speicherkapazitäten nicht ausreichen. Nicht praxisgerecht, das bedeutet: es fehlt ein Vorbild in der Praxis der Informationsverarbeitung und damit auch die Notwendigkeit, die fehlende Anwendungsmöglichkeit durch Programme zu unterstützen.

V. Individualität: Der Versuch von Beispielen

1. Wo ist Individualität möglich?

Gegeben sei die Aufgabe, Termine nach Datum, Uhrzeit, Ort und Gegenstand der Handlung, sowie die Namen der Gesprächspartner und das Ergebnis des Termins in Kurzform zu speichern, diese nach verschiedenen Kriterien wiederzugeben und, falls gewünscht, zu ändern oder zu löschen. Die bloße Erfüllung dieser Aufgabenstellung trägt noch keine individuellen Züge. Individuell könnte etwa eine durchgängige in Inhalt und Form erkennbare Erfolgsunterstützung in Planung und Durchführung von Terminen sein. Denkbar ist etwa aus dem Inhalt der gespeicherten Informationen an Auswertungen und Stützen, die erscheinen, ohne daß dazu separate Anweisungen gegeben werden. Die Vermeidung von Terminkollisionen und der Hinweis auf Wegezeiten reicht sicher nicht aus, markiert aber die Richtung. Memos während eines Termins zielgerichteter Art, eine programmierte Auswertung der Gesprächsergebnisse oder die Planung von Zeitkapazitäten gehen in dieselbe Richtung. Individuelle Erkenntnisse des Programmschöpfers können dem Benutzer Hilfe geben, die über banale Stützen hinausgehen und ein intellektuelles Additiv in der Programmstruktur darstellen.

Ein vielleicht anspruchsvolleres Beispiel kann für die Verarbeitung von Texten angedacht werden. Programme zur Textbe- und Textverarbeitung behandeln den eingegebenen Text ausschließlich formal. Darunter fällt auch die manchmal angebotene Hilfe bei der Trennung von Silben oder der orthographischen Prüfung. Ein derartiges Programm könnte jedoch auch so gestaltet sein, daß es den eingegebenen Text inhaltlich analysiert und so speichert, daß er auch unter anderen Merkmalen aufgefunden werden kann als denen, für die Speicherung angewiesen wurde. Die ergänzende Analyse ist automatisch durchzuführen, ohne besondere Anweisung des Autors, und zwar so, daß der Sprachschatz des Autors voll genutzt und erhalten bleibt, ohne daß etwa eine künstliche Begriffswelt geschaffen oder mit Abkürzungen gearbeitet wird.

Konkret: die Korrespondenz eines Autors mit A, die Sachverhalte zum Thema X beschreibt, könnte für den

Nutzer dann besonders wertvoll sein, wenn dieser bereits Gespeichertes zum Thema X sucht, das er aber nicht A, sondern B vermitteln will und nicht mehr weiß, daß er diesen Sachverhalt bereits unter der Kennung „A“ gespeichert hat.

Ein Programm zur Textverarbeitung könnte, ohne die heute gegebenen Möglichkeiten der Technik zu überreizen, in den erfaßten und gespeicherten Texten unterscheiden zwischen „geschriebenem“ und „gedachtem“ Text (= Gedankengut). Während ersteres auch gedruckt wird, bleibt das gedachte für fremde Leser „unsichtbar“, bietet aber dem Nutzer wertvolle Hilfen bei der Wiederverwendung der Texte. Koppelt man eine derartige Funktion mit der oben erwähnten Funktion der assoziativen Speicherung/Wiederauffindung, hätten wir eine große Bereicherung der heute angebotenen Textsysteme. Weder die eine noch die andere Funktion sind bis heute realisiert. In ihrer Gestaltung wäre — anders als bei der Fakturierung und ähnlichen Programmen — Raum für individuelle Lösungen und die Werkhöhe, die einen Schutz rechtfertigt.

Software ist dem Wesen nach am ehesten vergleichbar mit Musik. Da Musik umgekehrt wiederum sehr viel mit Mathematik zu tun hat, kommt der Vergleich nicht von ungefähr. Die Primitiv-Funktionen des Computers (Lesen, Schreiben, Speichern) bilden das Notensystem, das durch die Programmierwerkzeuge, die Programmiersprachen, für den Programmierer nutzbar gemacht wird. Der Programmierer ist also der Komponist. Im Prinzip.

Im Gegensatz zur Kunst der Musik, die dem Komponisten jeden Freiraum läßt, ist der Programmierer mit Sachzwängen behaftet, die ihn daran hindern, eigene „Kompositionen“ zu erzeugen, sondern ihm im wesentlichen genau vorschreiben, wie die Komposition zu klingen hat, welches Orchester (= Peripherie) ihm zur Verfügung steht, ja sogar, wie lange die einzelnen Töne (= Primitiv-Funktionen) dauern dürfen — letzteres nennt man performance, die zeitliche Nutzung der Ressourcen. Es gibt nur wenige Beispiele in der Software, in denen die Programmierer Freiraum hatten. Das bekannteste Beispiel ist vielleicht das Programm „LISA“ der Apple Inc.

Freilich: obige Hinweise sind nur Beispiele für Gebiete, wo die Schöpferkraft der Programmierer arbeiten könnte. Die Hürde, daß der BGH den schöpferischen Gehalt des technischen oder wissenschaftlichen Inhalts unbeachtet lassen will und nur die formale Gestaltung in individueller Prägung als schutzrechtsbegründend gelten lassen will¹⁴, ist damit noch nicht genommen. Schon oben wurde darauf hingewiesen, daß die Trennung zwischen Form und Inhalt sich praktisch nicht sauber durchführen läßt. Im Bereich der Software wird sie mit Sicherheit nicht befriedigen und, gedanklich verengt auf eine Linie, zur getarnten Willkür — man hätte dieses untaugliche Prinzip eigentlich überwunden geglaubt. Um mit Ulmer¹⁵ zu sprechen: auch bei wissenschaftlichen Werken ist der Inhalt nicht schlechthin frei und ungeschützt. Auch der BGH bemüht ja am Ende der Kriterien der Schutzwürdigkeit

verräterisch-verlegen den geistig-schöpferischen Gesamteindruck der konkreten Gestaltung¹⁶. In der Datenverarbeitung gibt es die Lehre, daß das Computerprogramm nichts anderes verkörpert als eine Äquivalenz zwischen Information und Steuerung¹⁷. Bei der Werkhöhe sollte wenigstens eine Äquivalenz der sogenannten inneren Form mit der äußeren auch für Software stattfinden. Begrifflich ist die Unterscheidung des BGH nicht nur unter Gerechtigkeitsgesichtspunkten, sondern auch dogmatisch dünnes Eis.

VI. Das den Durchschnitt Überragende

Das Können des durchschnittlichen Programmierers reicht nicht. Erst im „erheblich weiteren Abstand“ (BGH) der schöpferischen Eigenheiten zum handwerklichen Durchschnitt beginnt der Schutz. Der Durchschnitt muß deutlich überragt werden. Dies bereitet zusätzliche Probleme. Was nämlich für Software derzeit angestrebt wird, ist ein Stadium, in dem handwerklich saubere Software zum Durchschnitt wird. Man spricht von einer Softwarekrise, weil man sich von hohen Vergangenheitsinvestitionen in „schlecht konzipierte, strukturierte und dokumentierte“ Software gefangen fühlt¹⁸. Erfahrene Großanwender klagen, daß der Softwareentwicklungsprozeß nach wie vor unkontrollierbar und unvorhersehbar abläuft¹⁹. Offenkundig ist ein anerkanntes Software-Engineering, die gelegentlich angesprochenen „Grundsätze ordnungsgemäßer Programmierung“²⁰ noch nicht durchgesetzt. Die Qualität von Software wird derzeit in der Branche unter vielen Aspekten sehr breit erörtert. Ein handwerklich tragbarer Standard im Bereich einer dem Schutz zugänglichen Werkgattung wird jedoch in den Scheidungskriterien als Maßstab für eigenschöpferische Abweichungen vorausgesetzt — im Stand des Chaos sind nämlich Maßstäbe jeder Art untauglich. Auch wenn die urheberrechtliche Lehre sich darüber einig ist, daß eine künstlerische Bewertung nicht ihre Aufgabe zu sein hat, wenn sie Schutz und Schutzlosigkeit ortet, so braucht sie doch eine der Werkgattung innewohnende Regel, an der sie sich festhalten kann. | Die Tendenz der Branche geht dahin, den Programmierern derzeit statt der Pflege individueller Freiheiten systematisches, regelhaftes und qualitätsbewußtes Arbeiten beizubringen. Ist das nicht eine Aufforderung, unindividuelle und damit dem Schutze unzugängliche Programme zu schaffen? Auf den ersten

¹⁴ BGH GRUR 1984, S. 660

¹⁵ a. a. O., S. 121

¹⁶ BB 85, S. 1750, Sp. 2

¹⁷ zitiert bei Ganzhorn, Schulz, Walter, Datenverarbeitungssysteme, Berlin/Heidelberg/New York 1981, S. 5

¹⁸ Bender, Standard-Software: Silberstreifen oder Fata Morgana in Compas 85, Proceedings, Berlin 1985, S. 109

¹⁹ Hebestreit, Nutzenerwartungen eines Anwenders aus den Qualitätsanforderungen des Gütezeichens, in Compas 85, Proceedings, Berlin 1985, S. 1139

²⁰ Bartsch, Gewährleistungsrecht, Schadensersatzrecht und Recht der allgemeinen Geschäftsbedingungen für Software, in Wirtschaftsgut Software, Stuttgart 1985, S. 293

Blick womöglich. Man darf freilich den Faktor Zeit nicht außer acht lassen. Die gesamte Softwarebranche ist schließlich noch sehr jung; daß — bezogen auf die Programmierung überhaupt — erst Regeln entwickelt werden müssen und hinterher der Meister dann die Form zerbricht und Schutz erhält — all dieses braucht Zeit. Wollte man nämlich in zu frühem Stadium mit dem weitgehenden und langdauernden Schutz des Urheberrechts eingreifen und dabei womöglich diesen noch zu tief ansetzen, wäre eine Entwicklungsblockade möglich, die in niemandes Interesse sein kann. Ein solcher Gedanke ist nicht neu. Er hat ausgesprochenmaßen schon in den 60er Jahren ein Schutzrecht für Programmiermethoden in den USA behindert²¹. Obendrein muß man zusätzlich nach den Anwendungsarten der Programme differenzieren, da sich Einzelstandards (Allgemeingut) im Sinne des Urheberrechts bilden können.

VII. Zwischenergebnis

Die bisherigen Erwägungen nötigen zu dem Schluß, daß Programme, die nach den Regeln der derzeitigen Urheberrechtslehre und der höchstrichterlichen Rechtsprechung haltbaren Schutz genießen, zur Ausnahme gehören müßten — es sei denn, daß sich die rechtlichen Kriterien verändern oder die Software sich entwickelt. Die Branche muß sich jetzt darüber klar werden, ob sie einen zusätzlichen Schutz neben dem urheberrechtlichen überhaupt benötigt, ob ein solcher Schutz nicht vielmehr den notwendigen Reifungsprozeß der Gesamtsoftware verlangsamt: erst dann müßte sie sich in Richtung eines Sondergesetzes Gedanken machen. Entgegen Bauer²² ist die Bundesregierung nämlich allein durch den möglichen Ausfall des Urheberrechts noch zu gar nichts aufgerufen, solange die Branche ein Bedürfnis nicht dezidiert geäußert hat. Wie beim Know-how kann man den Schutz womöglich auch in der Praxis regeln.

VIII. Exkurs: Blick über die Grenzen

Da ein Argument für den Urheberschutz bei Software immer das Bestehen internationaler Konventionen war²³, die auch rasch supranational einen Schutz böten, ist es angezeigt, einen kurzen Blick auf das ausländische Urheberrecht zu werfen, ob unser Thema auch dort von Bedeutung ist. Schließlich sehen sowohl RBÜ (Art. 5) als auch WUA (Art. II) einen Schutz nach dem jeweiligen Inlandsrecht vor.

Die französische Rechtsprechung hat sich in der wichtigen Entscheidung *Apple ./ Segimex* vom 21. 09. 1983²⁴ um die Frage ausreichender Werkhöhe nicht besonders gekümmert. Man hat sich ziemlich pauschal damit zufriedengegeben, daß der Programmierer in seiner Persönlichkeit das Programm präge, wobei das „Tribunale de Grande Instance“ Paris interessanterweise genau jenes vom BGH verworfene Argument gelten läßt, schon bei einfachen Programmen böten verschiedene Programmierer bei derselben Auf-

gabenstellung eine Fülle von Variationen in der Ausführung. Offenkundig scheint in Frankreich keine derart umfassende Analyse wie hierzulande gefragt, um bei jedem Einzelwerk kritisch den Bestand an individuellen Zügen zu prüfen. Entscheidend ist nach den allgemeinen Lehren des französischen Urheberrechts die Originalität in der Form²⁵. Schon wenige Worte eines Werktitels haben für Originalität ausgereicht²⁶.

England hat 1985 auch die Computer-Software durch Gesetz in den Kreis der durch den Copyright-Act geschützten Werke aufgenommen²⁷. Zuvor gab es Rechtsprechung²⁸, die jedoch anscheinend keine Fragen der Werkhöhe, sondern allenfalls des Prinzips erörterte. Auch im englischen Recht ist die Originalität in der Form maßgeblich, aber es ist nur ein geringes Maß davon erforderlich, reduziert fast darauf, daß etwas von einem Autor herrührt (to originate)²⁹. In diesem Sinne scheinen Probleme bei der Werkhöhe nicht naheliegend, da jetzt das Prinzip des Schutzes vom Gesetzgeber anerkannt ist.

In den USA ist Computer-Software spätestens seit dem Amendment des Copyright-Act vom 12. 12. 1980 geschützt³⁰. Auch hier ist für den Schutz „Originalität“ erforderlich. Scott³¹ allerdings führt wörtlich aus: „Originalität ist ein viel geringerer Standard zur Erlangung des Schutzes als etwa die Erfordernisse von „Neuheit“ und „Nützlichkeit“ im Patentrecht. Nach Scott wäre bei dem Standard der Originalität praktisch jedes Computer-Programm originell, das nicht von einem anderen abkopiert ist. Ausnahmen sollen für sehr kurze Programme gelten und für solche, die sklavisch einem detaillierten Fluß-Diagramm folgen.

Dieser kurze Ausflug in das Recht von Staaten, die im Bereich der EDV-Märkte eine erhebliche Rolle spielen, läßt den vorläufigen Schluß zu, daß in den zitierten Ländern — und wohl auch in den weiteren Ländern des Common-Law-Rechtskreises — der Schutz der Software regelmäßig nicht an mangelnder Werkhöhe scheitern dürfte. Die strengen Anforderungen, die hierzulande gelten, werden andernorts wohl eher auf Verwunderung stoßen.

²¹ *Fisher, McGowan, Greenwood*, Der Anti-Trust-Fall US gegen IBM, Stuttgart 1985, S. 202

²² Rechtsschutz von Computerprogrammen in der Bundesrepublik Deutschland — eine Bestandsaufnahme nach dem Urteil des Bundesgerichtshofs vom 9. Mai 1985, CuR 1985, S. 12

²³ vgl. die Wiedergabe der Diskussion zum Thema bei *Schulze*, ZUM 1985, S. 35

²⁴ *Gazette du Palais*, 1984, Nr. 20 249

²⁵ *Debois*, Le Droit d'auteur en France, Paris 1978, S. 4 (2.)

²⁶ vgl. die ausführliche Darstellung bei *Laresse*, Urheberrecht in einem sich wandelnden Kulturbetrieb, Bern 1979, S. 183 ff.

²⁷ vgl. GRUR Int. 1985, S. 698

²⁸ referiert bei *August*, Computer program Protection: A world-view of developments, in „Computer Law and Practice“ 1985, S. 119

²⁹ vgl. *Copinger and Skone-James*, On Copyright, 11. Auflage 1971, RZ 112

³⁰ *Scott*, Computer Law, New York 1984, § 3.7

³¹ a. a. O., § 3.18

IX. Ein spekulativer Ausblick

Es besteht sicher Anlaß, darüber in Sorge zu sein, ob das Recht der Bundesrepublik künftig die Software — sei sie nun in ihrem heutigen Stadium der „Sophistication“ oder weiterentwickelt — ausreichend schützen wird. Andererseits — der Musterprozeß, in dem das höchstrichterliche Urteil erging, ist noch nicht entschieden. Es mag also zu einem weiteren Urteil des BGH kommen. Denkbar ist jedoch, daß in der rechtlichen Alltagspraxis die Anerkennung von Urheberrechten weitergehen wird, weil man dies für praktischer erachtet und das Risiko gerichtlicher Prüfung scheut. Bemerkenswert sind in diesem Zusammenhang die von Gravenreuth³² berichteten Entscheidungen im strafrechtlichen Bereich; dort wird anscheinend die Frage des Urheberrechtsschutzes wesentlich pauschaler im Einzelfall bejaht.

Denkbar wäre auch, daß im Rahmen einer weiteren Urheberrechtsänderung die Schutzanforderungen geringer gehalten werden, wobei als Ausgleich eine geringere Schutzdauer erwogen werden könnte. Vielleicht bringen uns auch europäische Einigungsbestrebungen einem adäquaten Schutz näher. Der unwahrscheinlichste aller Fälle dürfte jedoch ein an sich wünschenswerter sein: daß der Gesetzgeber einmal prinzipiell nachdenkt und im Rahmen einer BGB-Reform einen allgemeinen Teil und ein eigenes Buch für immaterielle Wirtschaftsgüter, darunter auch die Software, schafft.

³² Die Praxis der strafrechtlichen Verfolgung der Software-Piraterie, ZUM 1985, S. 539 (540)

Abkehr vom Urheberrechtsschutz für Computerprogramme?

Moritz Röttinger*

Im vergangenen Jahr gab es zwei bedeutende Ereignisse in der Bundesrepublik Deutschland, die den Rechtsschutz von Computerprogrammen betrafen. In einem Musterprozeß entschied der Bundesgerichtshof am 9. Mai 1985 erstmals über die Frage des Software-Schutzes, am 24. Juni 1985 wurde eine Novelle zum Urheberrechtsgesetz beschlossen. Im folgenden soll die Bedeutung dieser zwei Ereignisse für die Zukunft sowie der gegenseitige Einfluß beleuchtet werden.

I. Die BGH-Entscheidung

1. Dieser über mehrere Jahre dauernde Musterprozeß ist durch ein sehr genaues Vorgehen der Klägerin bei der Prozeßvorbereitung und -führung gekennzeichnet. Zur Beweissicherung wurde das Landeskriminalamt herangezogen, ein Gutachten über die Identität der Programme von Klägerin und Beklagten hatte große Bedeutung.

Die klagende Inkassogesellschaft ließ sich vom Beklagten ein Inkassoprogramm erstellen. Der Beklagte fertigte sich von diesem Programm Kopien an und lizenzierte diese in weiterer Folge an Dritte, was die Klägerin in diesem Rechtsstreit bekämpfte. Sie stützte sich, abgesehen von vertraglichen Anspruchsgrundlagen, die nur in diesem Einzelfall von — allerdings großer — Bedeutung waren und daher hier nur am Rande beleuchtet werden — auf Urheber- und Wettbewerbsrecht.

Das LG Mannheim fällt am 12. Juni 1981¹ das erstinstanzliche Urteil und war damit eines der ersten Gerichte, die sich mit dieser Problematik auseinanderzu-

setzen hatten². Dieses Urteil wirkte wie ein Paukenschlag³, denn es lehnte Urheberrechtsschutz kategorisch ab. Computerprogramme seien „einem urheberrechtlichen Schutz grundsätzlich schon von ihrem Gegenstand her nicht zugänglich“, weil es „regelmäßig ... an der erforderlichen, der sinnlichen Wahrnehmung zugänglichen Konkretisierung eines geistig-ästhetischen Gehalts“ fehle, so daß auf das Vorhandensein einer schöpferischen Leistung nicht geschlossen werden könne. Die Wertneutralität des Urheberrechts verkennend, vermißte das Gericht „das ästhetische Empfinden ansprechende Merkmale“ und forderte ei-

* Dr. jur., Wien—München, wissenschaftlicher Mitarbeiter am Max-Planck-Institut für ausländisches und internationales Patent-, Urheber- und Wettbewerbsrecht, München.

¹ BB 1981, 1546.

² Zwar entschied das LG Kassel am 21. Mai 1981 (BB 1983, 992), daß ein Computerprogramm urheberrechtlich schutzbar sei, und hielt eine 70 %ige Übereinstimmung zwischen dem Quellenprogramm des Klägers und des Beklagten für eine Urheberrechtsverletzung für ausreichend. Diese Entscheidung wurde vom OLG Frankfurt am 6. November 1984 (BB 1985, 139) bestätigt. Mit Beschluß vom 26. September 1985 (Aktz. I ZR 219/84) hat der BGH die Revision des Beklagten zur Verhandlung nicht angenommen, da die Rechtssache keine grundsätzliche Bedeutung und die Revision im Ergebnis keine Aussicht auf Erfolg habe. Zu beachten ist aber, daß dieser Beschluß nach dem Grundsatzurteil vom 9. Mai 1985 erfolgte. Doch ist es m. E. zweifelhaft, ob die Frage des Rechtsschutzes von Computersoftware aufgrund eines einzigen BGH-Urteils keine grundsätzliche Bedeutung mehr hat.

³ Vgl. Dietz, Das Problem des Rechtsschutzes von Computerprogrammen in Deutschland und Frankreich. Die kategoriale Herausforderung des Urheberrechts, BIE 1983, 305 (307).