

Seminar „Informatik und Recht“
der ETH und Universität Zürich
im Sommersemester 1995

Softwareschutz contra Softwarestandardisierung

Abgrenzung von Schutz- und Freihaltungsinteresse
bei EDV- und Software-Systemen

Beat Döbeli

7. Sem. Informatik ETHZ

Seminararbeit bei
Prof. Dr. C.A. Zehnder
Betreuer: Paul Trunz

0 Abstract

Bei Softwareentwicklung entsteht der Interessenkonflikt von Softwareschutz und Softwarestandardisierung.

Die vorliegende Arbeit versucht einerseits, die unterschiedlichen Interessen der Beteiligten abzugrenzen und andererseits die rechtliche Situation seit dem Inkrafttreten des revidierten Urheberrechtsgesetzes in der Schweiz aufzuzeigen.

Der Schwerpunkt liegt dabei auf einer verständlichen Einführung sowohl für Juristen als auch für Informatiker.

0.1 Zur Notation

Times New Roman 12p

Haupttext

Times New Roman 12p fett

Wichtige Stichworte

⁰ Times New Roman 10p

Fussnoten

Times New Roman 8p mit Rahmen

Gesetzestexte

Personenbezeichnungen beziehen sich immer auf Frauen und Männer.

Inhaltsverzeichnis

0 ABSTRACT	ii
0.1 Zur Notation	ii
1 EINLEITUNG	1
2 PROBLEMSTELLUNG UND INTERESSENKONFLIKTE	2
2.1 Das Schutzinteresse	2
2.2 Das Freihalteinteresse (Standardisierungsinteresse)	2
2.3 Beispiele von Standards:	4
3 INFORMATIKSPEZIFISCHE BEGRIFFE UND DEFINITIONEN	5
3.1 Software	5
3.2 Computerprogramm	5
3.3 Schnittstellen	6
4 ENTWICKLUNG DES SOFTWARESCHUTZES	8
4.1 Softwareschutz im Rahmen des URG oder Schutz sui generis ?	8
5 URHEBERRECHTLICHE GRUNDSÄTZE UND DIE PROBLEMATIK IHRER ANWENDUNG AUF COMPUTERPROGRAMME	9
5.1 Form geschützt, Inhalt frei	9
5.2 Der Werkbegriff	9
5.3 „Individueller Charakter“ eines Werkes	10
5.4 Art 2, Abs 4 URG „Ebenfalls geschützt sind [...] Teile von Werken [...]“	10
6 ABGRENZUNG VON SCHUTZ- UND FREIHALTEINTERESSE BEI VERSCHIEDENEN ASPEKTEN VON COMPUTERPROGRAMMEN	12
6.1 Vertikal: Abstraktionsgrad	12
6.2 Horizontal: Schnittstellen	13
7 THESEN	14
8 FOLIEN	16
9 QUELLENVERZEICHNIS	18
9.1 Gesetze, Verordnungen und Richtlinien	18
9.2 Literatur zum Softwareschutz in der Schweiz	18
9.3 Literatur zum Softwareschutz im Ausland	19
9.4 Weitere zitierte Stellen für allgemeine Definitionen	19

1 Einleitung

Das Thema „*Softwareschutz contra Softwarestandardisierung*“ betrifft die beiden Gebiete Informatik und Rechtswissenschaft. Als Informatiker musste ich zuerst die juristische Betrachtungsweise des Themas kennenlernen. Ähnlich wird sich wahrscheinlich ein Jurist mit den für ihn ungewohnten Objekten und Begriffen der Informatik vertraut machen müssen.

Als Fachmann nur des einen Gebietes wäre es mir nicht möglich gewesen, im Rahmen dieser Arbeit substantiell Neues zu diesem Thema zu entwickeln. Ich möchte aber auch nicht bereits Vorhandenes kopieren und bessere Abhandlungen schlecht wiedergeben.

Ziel meiner Arbeit ist vielmehr, eine auch für Nichtfachleute des einen oder anderen Gebietes verständliche Darstellung des Problematik und möglicher zukünftiger Entwicklungen zu geben.

So werde ich mich in dieser Arbeit auf die Situation in der Schweiz seit Inkrafttreten des revidierten Urheberrechtsgesetzes (rev. URG vom 1.7.93) beschränken und grösseres Gewicht auf die Erläuterung von Grundsätzen und Begriffen legen.

Für Interessierte sei auf das Quellenverzeichnis am Ende der Arbeit und insbesondere auf die Arbeit von Christoph Reuss [Reu93] verwiesen.

2 Problemstellung und Interessenkonflikte

Sowohl Softwarestandardisierung als auch Softwareschutz werden erst aktuell, wenn mehrere Parteien Software entwickeln. Es stellt sich bei der **Weiter-** oder bei der **Parallelentwicklung** von Software nämlich die Frage, welche Teile schützenswert sind und wo eine Standardisierung wünschenswert wäre.¹

Dieser Konflikt manifestiert sich einerseits als **Gegensatz zwischen technischen und wirtschaftlichen Interessen**: Vom rein technischen Standpunkt wäre eine **Standardisierung** und eine **Durchsetzung der besten Lösung** wünschenswert, vom wirtschaftlichem Standpunkt her ist ein gewisser **Leistungsschutz** unabdingbar.

Andererseits lässt sich dieser Konflikt auch als **Gegensatz der Interessen von Entwickler und Anwender** sehen: Während der Anwender vor allem ein Interesse an einer Standardisierung hat, wünscht sich der Entwickler einen Schutz der eigenen Arbeit.

2.1 Das Schutzinteresse²

Es ist eine Besonderheit von Software als Immaterialgut, dass sie sich mit sehr geringem Aufwand vervielfältigen lässt³.

Der technische Schutz von Software (**Kopierschutz**) hat verschiedene Nachteile:

- zusätzlicher Entwicklungsaufwand
- Störend für den legalen Benutzer⁴
- mögliche Fehlerquelle
- mit Aufwand trotzdem umgehbar

Für den Softwareentwickler ist somit ein wirksamer *rechtlicher* Schutz von Software von grossem Interesse. Primär geht es ihm dabei um den Schutz seiner materiellen (→ **Leistungsschutz**) und kreativen Arbeit (→ **Schutz von geistigem Eigentum**). Zudem muss bei vorhandenem Rechtsschutz die Entwicklung nicht geheimgehalten werden.

2.2 Das Freihalteinteresse (Standardisierungsinteresse)

Die Begründungen für eine Softwarestandardisierung sind vielfältig:

- Geringerer Entwicklungsaufwand: Bei der Verwendung von Standards kann viel Konzeptions- und Implementationsaufwand eingespart werden.
- Verbreitung der besten Lösung: Bei diesem Argument geht man davon aus, dass Software die Lösung eines technischen Problems ist, deren Güte sich objektiv messen lässt. Die beste Lösung soll sich durchsetzen und zu einem Standard

¹ Reines Kopieren von Software („Raubkopieren“) ist also **nicht** Thema dieser Arbeit.

² Folie 1 in Kapitel 8 dient als Illustration der folgenden beiden Kapitel

³ [Reu93]: „Bei anderen Immaterialgütern bestehen wenigstens noch faktische Hindernisse (Buch: Kopierkosten und -zeitaufwand, Handlichkeit Buch / Kopien; Musik: Abnehmende Tonqualität, Leerkassettenabgabe...“

⁴ Probleme bei der Erstellung von Sicherheitskopien, Wechsel der Hardware, Inkompatibilitäten der Schutzmassnahmen verschiedener Programme (Bsp. Dongles).

werden. In diesem Prozess wäre ein zu starker Softwareschutz störend.

- Das Rad nicht neu erfinden: Dieses Argument zielt in die gleiche Richtung: Grundlegende Probleme sollen nicht jedesmal neu gelöst werden müssen.

Während die bisherigen Argumente vor allem von Entwicklerseite stammen, sind die folgenden vor allem für den Anwender wichtig:

- Geringerer Lernaufwand: Der Lernaufwand für den Anwender ist um so geringer, je mehr eine neue Software dasselbe Modell der Wirklichkeitsrepräsentation, dieselben Metaphern und Elemente⁵ der Benutzerschnittstelle wie bereits bekannte Software verwendet.
- Zusammenarbeit von Software: Sowohl im Betrieb als auch bei der Ablösung von Software spielt die Zusammenarbeit von verschiedenen Programmen eine grosse Rolle. Standards helfen, diese Zusammenarbeit zu gewährleisten.
(im URG **Interoperabilität** genannt)

Bei real existierenden Standards auf dem Gebiet Software lassen sich in etwa drei Typen unterscheiden:

- Standards von Gremien: Solche Standards sind das Ergebnis von Standardisierungsgremien (Bsp. ISO: International Standards Organization), in denen interessierte Firmen und Organisationen vertreten sind. Da solche Standards zur allgemeinen Verwendung gedacht sind, ergeben sich keine urheberrechtlichen Probleme.
- de facto-Standards: Hierbei handelt es sich nicht um abgesprochene Übereinkünfte, sondern um durch wirtschaftliche oder technologische Vorsprünge entstandene Standards. Das Benutzen des Standards wird vom Entwickler je nach eigenen Interessen gefördert (Bsp. Microsoft Office) oder zumindest toleriert. Da solche de facto-Standards stark mit wirtschaftlichen Interessen gekoppelt sind, ist deren Zukunft und Weiterentwicklung oft unsicher. Probleme auch juristischer Art ergeben sich beispielsweise, wenn sich ein Entwickler eines Standards nicht mehr an die (ungeschriebenen) Regeln seines Standards hält um Konkurrenten zu benachteiligen.

⁵ Hierzu gehören Menüstruktur, graphische Oberfläche, Tastenkombinationen, Fehlermeldungen.

- Geschützte Standards von Firmen: Solche Standards dienen entweder der Förderung (Konsistenz, einheitliche Erscheinung) der eigenen Produkte oder wurden mit der Absicht geschaffen, mit Lizenzgebühren Geld zu verdienen.

2.3 Beispiele von Standards:

	Benutzer-schnittstelle	Daten-schnittstelle	Bibliotheks-schnittstelle
Gremienstandards:	X-Windows	RichTextFormat (RTF), ASCII	ISDN-CAPI ⁶
de facto-Standards:	„Microsoft Office compatible“ ⁷	dBase-Datenbankformat	Hayes Modem Befehle ⁸
Geschützte Standards:	Apple Macintosh-Oberfläche	PostScript	

⁷ Windows selbst ist kein de facto Standard: Programme die unter Windows ablaufen, erhalten zwar durch den Aufruf von Windows-Funktionen (als Client) eine Windows-Benutzerschnittstelle, Microsoft würde aber den Nachentwicklung eines Microsoft-Windows-Systems (als Server) sicher nicht tolerieren.

Bei „Microsoft Office compatible“ hingegen ist dies anders. Microsoft fördert die Verwendung derselben Objekte (Button-Leiste, Kontext-Menues) und Metaphern ihrer Office Produkte durch die Abgabe eines „Microsoft Office compatible“ Logos an Produkte anderer Hersteller, die sich gleich wie MS Office Produkte bedienen lassen. Dies stärkt natürlich die Bedeutung des Office Standards und nützt somit auch Microsoft.

⁸ Ursprünglich nur für die Steuerung von Modems (also von Hardware) gedacht, werden Hayes-Befehle nun auch für die Steuerung anderer Kommunikationsverbindungen verwendet.

3 Informatikspezifische Begriffe und Definitionen

Die mit neuartigen Immaterialgüterobjekten an sich schon diffizile Erörterung von Rechtsfragen in der Informatik wird zusätzlich erschwert durch die unsaubere Verwendung von Fachbegriffen:

„Die Begriffe *Software* und *Computerprogramm* werden allerdings aus praktischen Gründen im folgenden auch synonym verwendet.“⁹

„In der französischen Sprache scheint zudem der Gesetzgeber zwischen „Programmes d’ordinateurs“ und „Logiciels“ keinen Unterschied zu machen, während in der deutschen Sprache in der Regel zwischen Software und Computerprogrammen unterschieden wird.“¹⁰

Deshalb möchte ich für das Verständnis des Folgenden zuerst einige Fachbegriffe definieren und erläutern.

3.1 Software

In dieser Arbeit gehe ich von folgender Definition von Software aus:

Software: Alles was für Lösung eines Problems mit Computerhilfe nötig ist, abzüglich der Computerhardware (d.h. ohne die materiellen Bestandteile).
Zur Software zählen also insbesondere:

- das Computerprogramm
- die verwendeten und die entstehenden Daten
- die Entwicklungsdokumentation (dieses Computerprogramms)
- die Anwenderdokumentation

3.2 Computerprogramm

DIN 44.300: „Ein **Computerprogramm** ist 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.“¹¹

Rauber: „Ein **Computerprogramm** ist „eine in sich geschlossene, logisch eindeutige Befehlsfolge, welche bezweckt und nach Eingabe in einen Computer unmittelbar bewirkt, dass dieser Computer ihm vom Benutzer eingegebene Daten nach einer bestimmten Aufgabenstellung (Applikation) verarbeitet und das Verarbeitungsergebnis nach Bedarf anzeigt oder ausgibt.“¹²

EG-Richtlinie: „Für die Zwecke dieser Richtlinie soll der Begriff „**Computerprogramm**“ Programme in jeder Form umfassen, auch solche, die in die Hardware integriert sind; dieser Begriff umfasst auch Entwurfsmaterial zu

⁹ [Tho92] Fussnote 3 auf Seite 1

¹⁰ [Lutz93]

¹¹ Zitiert nach [Reu93] S. 13

¹² [Rau88], zitiert nach [Reu93]

Entwicklung eines Computerprogramms, sofern die Art der vorbereitenden Arbeit die spätere Entstehung eines Computerprogramms zulässt.“¹³

3.2.1 Quellcode/Sourcecode und Maschinencode/Objektcode

Ein **Computerprogramm** wird von einem Menschen in einer Computersprache (Pascal, C usw.) geschrieben. Diese vom Menschen lesbare Form eines Programmes wird **Quellcode** oder **Sourcecode** genannt. In dieser Form ist das Computerprogramm zwar vollständig definiert, aber noch nicht auf einen bestimmten Computer begrenzt.

Der Sourcecode wird meist mit Computerhilfe übersetzt (= **compiliert**) in eine vom Computer direkt ausführbare Form, dem sogenannten **Maschinencode** oder **Objektcode**. Dieser Code kann vom Menschen nicht mehr direkt gelesen werden und ist nur auf einem bestimmten Computertyp lauffähig.

3.2.2 Algorithmus

Im Duden von 1985 ist der Begriff Algorithmus nicht computerspezifisch definiert:

Fremdwörterduden: [arab.-mlat] der; -, ...men: 1. (veraltet) Rechenart mit Dezimalzahlen. 2. Rechenvorgang, der nach einem bestimmten [sich wiederholenden] Schema abläuft (Arithmetik) 3. Verfahren zur schrittweisen Umformung von Zeichenreihen (math. Logik)¹⁴

Botschaft zum URG: „Ein Algorithmus ist ein vollständiger Satz wohldefinierter Regeln zur Lösung eines Problems in einer endlichen Zahl von Schritten.“¹⁵

3.3 Schnittstellen

Als **Schnittstellen** (engl. **Interface**) werden in der Informatik Teile eines Systems bezeichnet, welche in irgend einer Art und Weise mit anderen Systemen kommunizieren. Dies reicht vom besonders illustrativen Beispiel von Steckern und Steckdosen bis zu abstrakteren Schnittstellen wie die im Folgenden beschriebenen Benutzer- oder Bibliotheksschnittstellen. Schnittstellen spielen beim Thema Softwareschutz als auch bei der Softwarestandardisierung eine zentrale Rolle:

- Einerseits sind Schnittstellen die erkennbaren Teile der „Blackbox Software“, die grosse Teile ihrer Eigenschaften definieren und somit eventuell schützenswert sind.
- Andererseits betreffen Standards meistens Schnittstellen, um die problemlose Zusammenarbeit mehrerer Systeme zu gewährleisten.

3.3.1 Die Benutzerschnittstelle - Das User-Interface

Das User-Interface ist die Schnittstelle der Systeme Mensch und Maschine. Dazu gehört alles, was für den Benutzer eines Computerprogrammes sichtbar wird: Bildschirmaufbau, Menüstruktur, besondere Tastenkombinationen, Hilfs- und Fehlermeldungen aber im weiteren Sinne auch Ausdrücke oder ganze Arbeitsabläufe.

3.3.2 Die Bibliotheksschnittstelle

Die **Bibliotheksschnittstelle** definiert die Kommunikation zwischen verschiedenen Programmen oder Programmteilen. So können Programme, die häufig auftretende Probleme lösen, (Bsp. Suchen, Sortieren oder Anzeigen) wiederverwendet werden, indem sie zu einer

¹³ Zitat [EG-Richtlinie], Einleitung

¹⁴ Fremdwörterduden, 1984

¹⁵ ISO 2382/1 von 1974, zitiert nach [Reu93]

Bibliothek zusammengefasst und von anderen Programmen aufgerufen werden. Auch hier ist natürlich eine Standardisierung solcher Schnittstellen wünschenswert, andererseits stellt die Entwicklung solcher Schnittstellen auch eine Leistung dar, die eventuell geschützt werden soll.

4 Entwicklung des Softwareschutzes

In diesem Kapitel soll aufgezeigt werden, wie das in der Einleitung ausgewiesene Schutzbedürfnis in der Schweiz im Gesetz verankert wurde.

4.1 Softwareschutz im Rahmen des URG oder Schutz sui generis ?

Es würde den Rahmen dieser Arbeit sprengen, die Entwicklung der Rechtsauffassungen zum Thema Softwareschutz detailliert aufzuzeigen. Für das weitere Verständnis möchte ich die wichtigsten Punkte kurz antönen und ansonsten auf ausführlichere Literatur verweisen.¹⁶

Im ersten Urheberrechtsgesetz vom 1.1.1901 wird der Softwareschutz verständlicherweise nicht erwähnt. Dieses Problem stellte sich erst mit der Verbreitung von Computern in den 70-Jahren.

Die **Grundsatzfrage** der Rechtsdogmatiker lautete: **Wo soll der Softwareschutz im Gesetz verankert werden ?**

- Eine Möglichkeit besteht darin, den **Softwareschutz ins Urheberrecht zu integrieren**, indem man entweder Computerprogramme als neue Werkkategorie aufführt oder sie als literarische Werke auffasst.
- Eine andere Möglichkeit wäre, für Software ein **Sonderschutzrecht sui generis**¹⁷ zu entwickeln, wie dies z.B. bei integrierten Schaltungen („Chips“) erfolgt ist¹⁸.

Die internationale Rechtsentwicklung, der auch die Schweiz gefolgt ist, bevorzugt die erste Variante, der Einbettung des Softwareschutzes ins Urheberrecht.¹⁹ Gründe dafür waren vor allem die internationale Rechtsentwicklung und die Möglichkeit, im Rahmen der Revision des URG relativ rasch einen Softwareschutz einzuführen.

¹⁶ USA: [Oman92]

CH: [Reu93] 2.1 Zur Vorgeschichte des revidierten URG

¹⁷ sui generis: (lat.) durch sich selbst eine Klasse bildend

¹⁸ USA: Semiconductor Chip Protection Act 1984, Umfang und Motivation erläutert in [Oman92]

CH: Topographengesetz, ToG, 1993

¹⁹ Für Begründungen und Kritik dieser Entscheidung siehe

[Reu93], 1.3 Pro/ Contra Sonderschutzrecht für Programme

3.1a Kritik: Einordnung des Programmschutzes ins Urheberrecht

[Lutz93], 1. Die Problematik der Computerprogramme

5 Urheberrechtliche Grundsätze und die Problematik ihrer Anwendung auf Computerprogramme

Im folgenden möchte ich einige für Computerprogramme relevante urheberrechtliche Grundsätze diskutieren. Die Anwendung dieser Grundsätze auf Computerprogramme ist nicht immer einfach und wird oft durch fehlende Kenntnisse des jeweilig anderen Fachgebietes bei Juristen und Informatikern erschwert.

5.1 Form geschützt, Inhalt frei

Ein wichtiger urheberrechtlicher Grundsatz besagt:

Die Form eines Werkes ist geschützt, nicht aber dessen Inhalt.²⁰

Somit stellt sich die (nicht nur bei Computerprogrammen vorhandene) **Problematik der Trennung von Form und Inhalt**. Die Meinungen zu diesem Thema in Bezug auf Computerprogramme sind zahlreich; eine gute Übersicht gibt [Reu93] im Kapitel 14.3 *Form/Inhalt-Dilemma*.

Als Zusammenfassung lässt sich sagen, dass Programmideen und Algorithmen als nicht schützbar angesehen werden, Quellcode und Maschinencode hingegen urheberrechtlich geschützt sind.

5.2 Der Werkbegriff

In Art 2 URG wird ein Werk folgendermassen definiert:

URG Art. 2 Werkbegriff

- 1 Werke sind, unabhängig von ihrem Wert oder Zweck, geistige Schöpfungen der Literatur und Kunst, die individuellen Charakter haben.
- 2 Dazu gehören insbesondere:
 - a. literarische, wissenschaftliche und andere Sprachwerke;
 - b. Werke der Musik und andere akustische Werke;
 - c. Werke der bildenden Kunst, insbesondere der Malerei, der Bildhauerei und der Graphik;
 - d. Werke mit wissenschaftlichem oder technischem Inhalt wie Zeichnungen, Pläne, Karten oder plastische Darstellungen;
 - e. Werke der Baukunst;
 - f. Werke der angewandten Kunst;
 - g. fotografische, filmische und andere visuelle oder audiovisuelle Werke;
 - h. choreographische Werke und Pantomimen.
- 3 Als Werke gelten auch Computerprogramme.
- 4 Ebenfalls geschützt sind Entwürfe, Titel und Teile von Werken, sofern es sich um geistige Schöpfungen mit individuellem Charakter handelt.

Bereits Art 2 Abs 3 URG zeigt die problematische Einbettung von Computerprogrammen im Urheberrecht deutlich: Computerprogramme *gelten* als Werke, sind es also nicht wirklich.²¹

²⁰ Dieser Grundsatz ist **nicht** im URG ausformuliert. Auch in der [EG-Richtlinie] wird er nur in der Einleitung erwähnt: „Zur Vermeidung von Zweifeln muss klargestellt werden, dass der Rechtsschutz nur für die Ausdrucksform eines Computerprogramms gilt und dass die Ideen und Grundsätze, die irgendeinem Element des Programms einschliesslich seiner Schnittstellen zugrunde liegen, im Rahmen dieser Richtlinie nicht urheberrechtlich geschützt sind.“ und „Entsprechend diesem Urheberrechtsgrundsatz sind Ideen und Grundsätze, die der Logik, den Algorithmen und den Programm Sprachen zugrunde liegen, im Rahmen dieser Richtlinie nicht urheberrechtlich geschützt.“

²¹ Laut [Lutz93] stammt die Formulierung *gelten* aus dem Ständerat (Sten Bull. SR 1991, S. 98f. („Art 2^{bis}“))

Computerprogramme bilden im schweizerischen Urheberrecht somit eine eigene Werkkategorie, dies im Gegensatz zur EG-Richtlinie²² und zum amerikanischen Recht²³, welche Computerprogramme als literarische Werke auffassen.

5.3 „*Individueller Charakter*“ eines Werkes

In Art 2 Abs 1 und 4 URG taucht der Begriff des *individuellen Charakters* eines Werkes als Voraussetzung für dessen Schutz auf. Wie ist der *individuelle Charakter eines Werkes* definiert? Auch diese nicht nur für Computerprogramme wichtige Frage wird nicht im Gesetz beantwortet, sondern ist ein heftig diskutiertes Thema der Rechtsdogmatik. Für eine detaillierte Behandlung der Thematik möchte ich wiederum auf die Arbeit von Reuss ([Reu93] 1.4.2 *Individualität/Originalität*) verweisen und hier nur eine Übersicht der wichtigsten Punkte geben.

Eine für Computerprogramme meiner Ansicht nach brauchbare Definition gibt laut [Reu93] Kummer, indem er Individualität mit **statistischer Einmaligkeit** gleichsetzt:

„Statistische Einmaligkeit meint, dass die Wahrscheinlichkeit, dass ein anderer Urheber (unabhängig) genau das selbe Werk wie ein anderer schafft, *praktisch* gleich Null ist.“²⁴

Der individuelle Charakter setzt also einen **Gestaltungsspielraum** voraus. Bei Computerprogrammen als Lösungen für technische Probleme stellt sich natürlich einerseits die Frage, ob dieser Gestaltungsspielraum überhaupt besteht und andererseits, ob es sinnvoll ist, ihn als Mass für die Schutzwürdigkeit zu nehmen.

Ein Detailproblem ergibt sich bei der Beurteilung von neuen Programmversionen, welche sich nur zum Teil von der Vorgängerversion unterscheiden, jedoch trotzdem Urheberrechtsschutz als eigenständiges Werk beanspruchen.²⁵

5.4 Art 2, Abs 4 URG „*Ebenfalls geschützt sind [...] Teile von Werken [...]*“

Dieser Absatz ist zu beurteilen unter Berücksichtigung des vorhergehenden Abschnittes. Je kleiner ein Teil eines Computerprogrammes ist, desto geringer ist natürlich auch der Gestaltungsspielraum. Ähnlich wie bei Literatur stellt sich zum Beispiel die Frage, wann eine bestimmte Formulierung eine syntaktische Notwendigkeit darstellt. Weder das Gesetz noch die Rechtslehre vermochte hier bisher eine scharfe Grenze zu ziehen.

Im Unterschied zu literarischen Kopien von Teilwerken, wo es sich entweder um Zufall oder um bewusstes Zitieren aus künstlerischen Überlegungen handelt, stecken beim Kopieren von Programmteilen handfeste wirtschaftliche Interessen dahinter: Solche „Bauteile“ lassen sich ohne erneute Entwicklungskosten zu neuen Programmen zusammenstellen.

Im Gegensatz zu Literatur werden Programme oft mit expliziter Berücksichtigung ihrer **Wiederverwertbarkeit** geschrieben.

²² [EG-Richtlinie] Art. 1, Abs. 1: „Gemäss den Bestimmungen dieser Richtlinie schützen die Mitgliedstaaten Computerprogramme urheberrechtlich als literarische Werke im Sinne der Berner Übereinkunft zum Schutze von Werken der Literatur und der Kunst.“

²³ [Oman92] Zitat: „Das U.S. Copyright Office registriert Computerprogramme seit dem Jahre 1964. Dabei ging es um die rechtliche Frage, ob Programme ein „Schriftwerk“ im Sinne des Gesetzes darstellten oder nicht. Wir gelangten zu dem Schluss, dass es sich um Sprachwerke handelte, und dass wir ein Programm als „Buch“ registrieren konnten.“

²⁴ [Reu93] Seite 9

²⁵ Siehe dazu [Tho92] S.31 und [Reu93] S.9

Auch hier zeigt sich somit der grundsätzliche Unterschied zwischen Computerprogrammen und anderen *künstlerischen* Immaterialgüterkategorien.

6 Abgrenzung von Schutz- und Freihalteinteresse bei verschiedenen Aspekten von Computerprogrammen

Ein Computerprogramm lässt sich in verschiedene Aspekte unterteilen.²⁶ Eine Möglichkeit besteht darin, den Abstraktionsgrad als Kriterium der Aufteilung zu wählen: Zuerst liegt eine Lösungsidee vor, dann ein von einer Programmiersprache unabhängiger Algorithmus, die Implementation mit einer bestimmten Sprache liegt zuerst als Quellcode vor und schliesslich nach der Übersetzung in maschinenlesbarem Objectcode.

Eine andere Möglichkeit der Aufteilung ist die Betrachtung der Schnittstellen eines Programmes als Grenzflächen, wo ein Programm Information mit der Umwelt austauscht und somit auch erfassbar wird.

Diese beiden Dimensionen dieser Aufteilung lassen sich nun zu einer Matrix vereinen, die deutlich zeigt, dass jeder Teil eines Programmes mehrerer Betrachtungsaspekte vereint:

	Benutzer- schnittstelle	Daten- schnittstelle	Bibliotheks- schnittstelle
↑ Lösungsidee			
Algorithmus			
Sourcecode			
Objectcode			

Ab-
straktionsgrad

Schnittstellen

Anhand dieser beiden Dimensionen der Aufteilung möchte ich im folgenden die Abgrenzung von Schutz- und Freihalteinteresse zeigen.

6.1 Vertikal: Abstraktionsgrad

Gemäss dem Grundsatz *Form geschützt, Inhalt frei* (Kapitel 5.1) fällt die Grenze der Schutzwürdigkeit in der Rechtslehre relativ eindeutig: Lösungsidee und Algorithmus sind (als Inhalt eines Computerprogramms) nicht schützbar, hingegen sind Source-Code und Object-Code (als Form des Programms) geschützt.

Diese Grenzziehung wird nicht nur in der schweizerischen Rechtslehre vertreten, sondern findet sich auch in der europäischen und amerikanischen Literatur.²⁷

Ob diese klare Trennung auch in der Rechtssprechung Anwendung findet, lässt sich noch nicht sagen, da seit Inkrafttreten des revidierten URG noch keine entsprechenden Urteile des Bundesgerichts gefallen sind.

Für die Standardisierungsbemühungen stellt diese Aufteilung kein Hindernis dar (Konzepte sind nicht geschützt), die Problematik beginnt erst bei den Schnittstellen:

²⁶ Eine Übersicht dieser Unterteilung und der entsprechenden Ergebnisse bietet Folie 4 in Kapitel 8

²⁷ [Oman92]

6.2 Horizontal: Schnittstellen

6.2.1 Benutzerschnittstelle

Die Benutzerschnittstelle ist urheberrechtlich geschützt, sofern sie den nötigen **individuellen Charakter** (Kapitel 5.3) aufweist. Das Problem entstände nun genau da, diese Individualität festzustellen und abzugrenzen, doch in der Praxis spielt dies keine Rolle mehr: Heutzutage werden weltweit keine entsprechenden Urteile mehr gefällt und einzig Lotus war je mit einer Klage wegen Kopie der Benutzeroberfläche erfolgreich.

Bei Standardsoftware hat sich die Benutzeroberfläche weitgehend vereinheitlicht, wichtig sind die (nicht schützbaeren) verwendeten Metaphern bei der Bedienung, nicht die genaue Umsetzung.

Einzig bei Branchensoftware, wo die Umsetzung einer Branchenproblematik in einen Arbeitsablauf (Bsp. Menüstruktur) individuellen Charakter ermöglicht, wäre die Anklage gegen Nachahmer denkbar.²⁸

Aus Anwendersicht ist der Wunsch nach Standardisierung eindeutig: Die Bedienung wird einheitlich und der Lernaufwand bedeutend geringer.

6.2.2 Datenschnittstelle

Auch die Datenschnittstelle (d.h. Dateiformat, Feldformat, Zeichensatz usw.) ist bei vorhandenem **individuellem Charakter** (Kapitel 5.3) im Prinzip geschützt. Als Einschränkung dieses Schutzes gilt es jedoch Art 21 URG zu beachten:

URG Art. 21 Entschlüsselung von Computerprogrammen

- 1 Wer das Recht hat, ein Computerprogramm zu gebrauchen, darf sich die erforderlichen Informationen über Schnittstellen zu unabhängig entwickelten Programmen durch Entschlüsselung des Programmcodes beschaffen oder durch Drittpersonen beschaffen lassen.
- 2 Die durch Entschlüsselung des Programmcodes gewonnenen Schnittstelleninformationen dürfen nur zur Entwicklung, Wartung sowie zum Gebrauch von interoperablen Computerprogrammen verwendet werden, soweit dadurch weder die normale Auswertung des Programms noch die rechtmässigen Interessen der Rechtsinhaber und -inhaberinnen unzumutbar beeinträchtigt werden.

Erklärung von Art 21 URG in Art 17 URV:

URV Art. 17 Schutz von Computerprogrammen

- 1 [...]
- 2 Nach Artikel 21 Absatz 1 URG erforderliche Informationen über Schnittstellen sind solche, die zur Herstellung der Interoperabilität eines unabhängig geschaffenen Programms mit anderen Programmen unerlässlich und dem Benutzer oder der Benutzerin von Programmen nicht ohne weiteres zugänglich sind.
- 3 Eine unzumutbare Beeinträchtigung der normalen Auswertung des Programms im Sinne von Artikel 21 Absatz 2 URG liegt insbesondere vor, wenn die im Rahmen der Entschlüsselung gewonnenen Schnittstelleninformationen für die Entwicklung, Herstellung oder Vermarktung eines Programms mit im wesentlichen ähnlicher Ausdrucksform verwendet werden.

Dieser Artikel sichert also die mögliche Gewährleistung der **Interoperabilität**: Datenschnittstellen dürfen verwendet werden, sofern kein Konkurrenzprodukt entwickelt wird.²⁹

²⁸ Mögliches Szenario: Eine Firma entwickelt eine Software für Gemeindeverwaltungen und investiert viel Arbeit in die möglichst zweckmässige Umsetzung der anfallenden Arbeitsabläufe in Menüaufbau und Programmablauf. Die Software findet grössere Verbreitung in Gemeindeverwaltungen, sodass eine andere Firma ein Konkurrenzprodukt entwickelt, welches sich gleich bedienen lässt. Ist die Verwendung der identischen Benutzeroberfläche nun erlaubt, weil die Aufgabenstellung keine grosse Individualität zulässt, oder handelt es sich um unerlaubte Verwendung von Teilen eines urheberrechtlich geschützten Werkes?

²⁹ Bsp. Textdateiformat für Import in Desktop-Publishing Programm.

Die Absicht des Gesetzgebers, den Benutzer nicht mit dem Urheberrechtsgesetz an ein einmal verwendetes Programm zu binden, leitet sich aus den Tatsachen ab, dass

- Daten wichtiger (=wertvoller) sind als Programme
- und Daten eine grössere Lebensdauer haben als Programme.

Das Urheberrechtsgesetz soll zwar die Leistung des Entwicklers schützen, aber andererseits nicht die berechtigten Möglichkeiten des Benutzers einschränken.

Somit ist eine Standardisierung in vertikaler Richtung (Datenfluss) möglich, die Entwicklung eines Konkurrenzproduktes (horizontale Standardisierung) mit gleicher Datenschnittstelle liesse sich jedoch mit dem URG einklagen.

Auch hierzu fehlen bisher Entscheide des Bundesgerichts, so dass sich erst noch zeigen muss, ob die Rechtssprechung dieser Argumentation folgt.

6.2.3 Bibliotheksschnittstelle

Bei der Bibliotheksschnittstelle liesse sich ähnlich wie bei der Datenschnittstelle argumentieren (im Prinzip geschützt, aber unter dem Vorbehalt der Gewährleistung der Interoperabilität), doch stellt sich die Frage, ob der Gesetzgeber den Aspekt von Bibliotheksschnittstellen bei der Entwicklung des revidierten URG überhaupt berücksichtigt hat.

Dieser Gedanke führt über zu meinen Schlussfolgerungen:

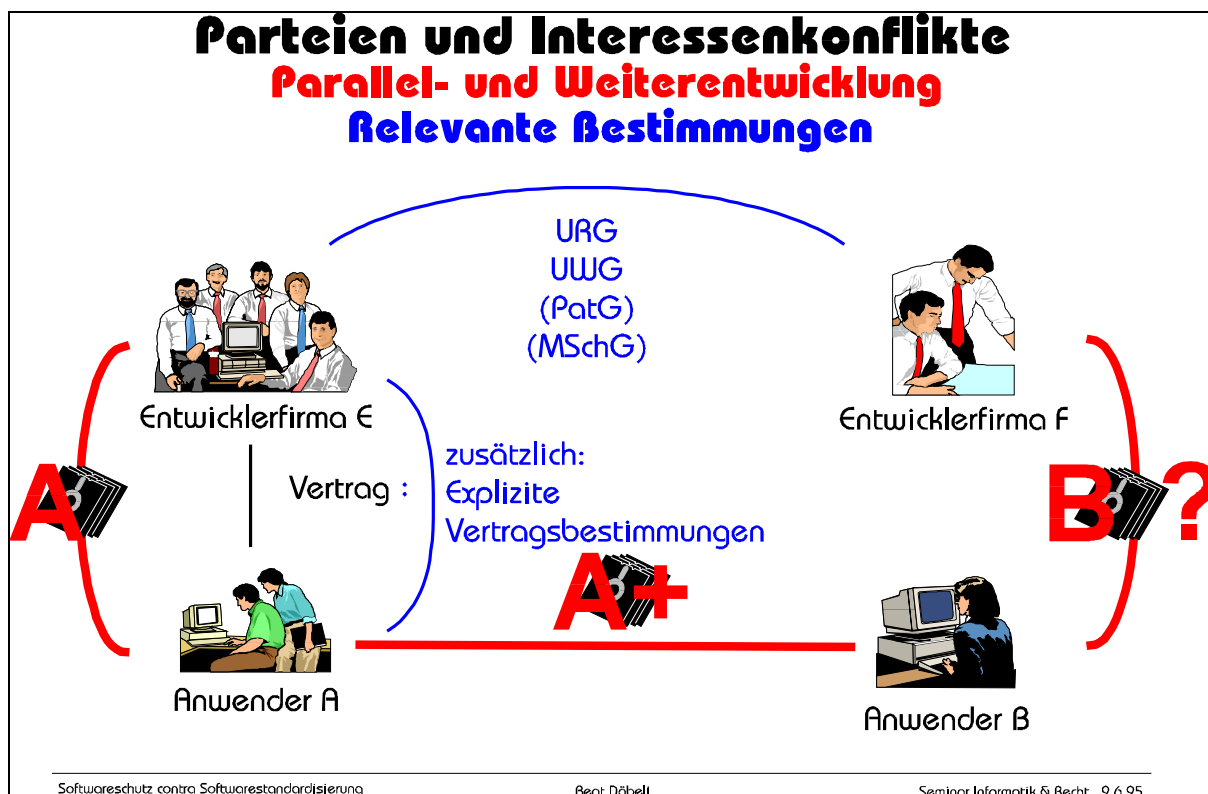
7 Thesen

- Auf dem Gebiet des Softwareschutzes ist die **Rechtssprechung wichtiger** als die **Gesetzgebung**: Im sich schnell wandelnden Gebiet der Informatik hinkt die Gesetzgebung immer der Entwicklung hinterher. Es ist somit Aufgabe der Rechtssprechung im Einzelfall die Absicht des Gesetzgebers auf die momentane Situation umzusetzen. Dies setzt jedoch die Kenntnis von Recht und Informatik sowohl beim Richter als auch bei Experten voraus, um sinnvolle und konsequente Lösungen finden zu können.
- Die Integration des Softwareschutzes in das Urheberrecht ist nachvollziehbar als Anpassung an die internationale Rechtsentwicklung und schnellstmögliche Herstellung irgendeines Schutzes von Software. Die **internationale Rechtsentwicklung** hingegen scheint mir **fragwürdig**: Programme als technische Lösungen von technischen Problemen verlangen Leistungsschutz, das Urheberrechtsgesetz hingegen schützt die künstlerische Leistung. Somit wird bei Computerprogrammen eine an sich technisch unerwünschte Individualität geschützt, die technisch beste Lösung hingegen bleibt schutzlos. Es stellt sich die Frage, ob nicht eine eventuelle Ausdehnung des Patentschutzes auf Computerprogramme diesbezüglich Vorteile brächte.³⁰
- Eine mögliche Entwicklung sehe ich darin, dass Software immer weniger als künstlerisches und somit schützenswertes Werk betrachtet wird, sondern als Produkt einer Ingenieurleistung, auf deren Kopie vor allem das Gesetz des unlauteren Wettbewerbs angewendet werden könnte.

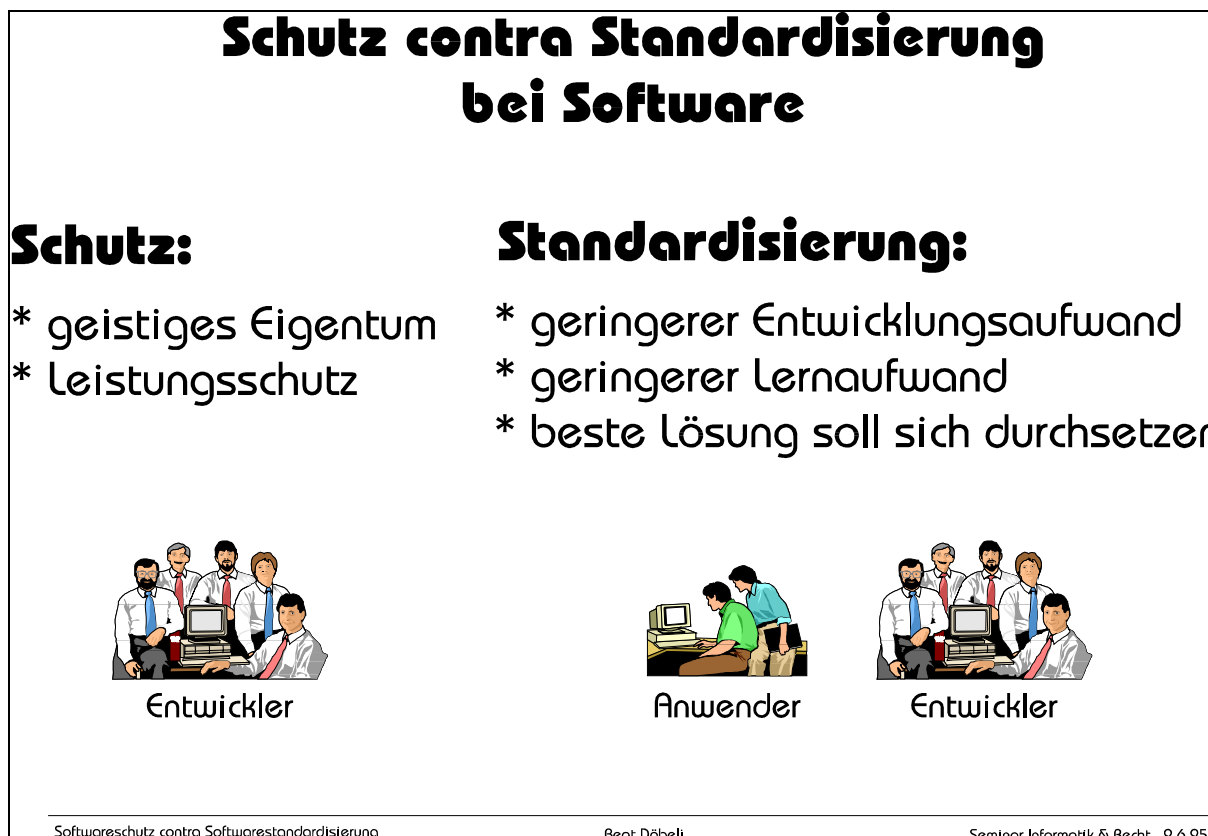
³⁰ Der Patentschutz hat gegenüber dem Urheberrechtsschutz den grossen Nachteil, dass er zuerst beantragt und erteilt werden muss und zudem a priori Kosten verursacht.

Dies könnte die Entwicklung von Standards fördern, auch wenn ich persönlich nicht das Gefühl habe, dass Gesetze bisher das grösste Hindernis bei der Durchsetzung von Standards darstellen.

8 Folien

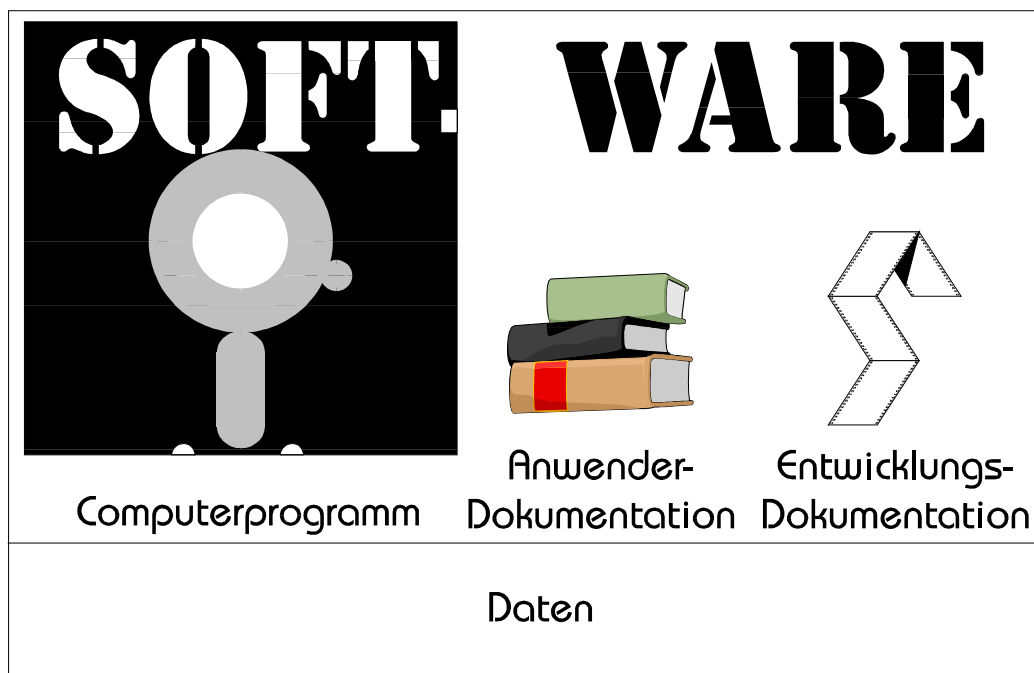


Folie 1: Parteien und Interessenkonflikte



Folie 2: Argumente Schutz contra Standardisierung

Software



Softwareschutz contra Softwarestandardisierung

Beat Döbeli

Seminar Informatik & Recht 9.6.95

Folie 3: Software

		Computerprogramm			
		Benutzer-schnitt-stelle	Daten-schnitt-stelle	Bibliotheks-schnitt-stelle	
Abstraktionsgrad	Programm-Idee				
	Algorithmus	Menüstruktur, Bildschirmaufbau, Hilfe, Meldungen	Dateiformat, Feldformat, Zeichensatz	API: Prozedur- & Funktionsaufrufe Bsp. OLE, OpenStep, Posix	Elemente, Bsp
	Quell-Code	sofern individuell	Nein, Art 21 URG: Gewährleistung der Interoperabilität	Nein, Art 21 URG: Gewährleistung der Interoperabilität	Geschützt ?
	Object-Code	geringerer Lernaufwand praktisch keine Urteile mehr, Lotus als einzige erfolgreich	geringerer Migrationsaufwand	geringerer Entwicklungsaufwand	Sinn einer Standardisierung Rechtsprechung

Softwareschutz contra Softwarestandardisierung

Beat Döbeli

Seminar Informatik & Recht 9.6.95

Folie 4: Schutz von Computerprogrammen

9 Quellenverzeichnis

9.1 Gesetze, Verordnungen und Richtlinien

- [CD-ROM] **Schweizer Gesetzestexte auf CD-ROM**
optobyte 1995
- [I&R] **Informatik und Recht**, wichtige Rechtsquellen, Literaturangaben zur
Vorlesung Informatik & Recht, P. Forstmoser, Andras Gurovits, Beat
Lehmann,
Universität Zürich, WS 94/95
- [URG] **BG über das Urheberrecht und verwandte Schutzrechte**
(Urheberrechtsgesetz), vom 9.10.92, in Kraft seit 1.7.93
Fundstellen: (SR 231.1), [I&R], [CD-ROM]
- [URV] **Verordnung über das Urheberrecht und verwandte Schutzrechte**
(Urheberrechtsverordnung)
Fundstellen: (SR 231.11), [I&R], [CD-ROM]
- [EG-Richtlinie] **Richtlinie des Rates vom 14. Mai 1991 über den Rechtsschutz von**
Computerprogrammen (91/250/EWG)
Fundstellen: [I&R]
 Amtsblatt der Europäischen Gemeinschaft, 17.5.91, Nr. L
 122/42
 c't 12/92, S. 103
- [D-URG] **Zweites Gesetz zur Änderung des Urheberrechtsgesetzes vom 9. Juni 1993**
Fundstellen: Bundesgesetzblatt 1993, Teil I, S. 910
 c't 12/93

9.2 Literatur zum Softwareschutz in der Schweiz

- [Tho92] Felix H. Thomann:
Grundriss des Softwareschutzes
Reihe Computer und Recht, Band 24,
Hrsg. K. Bauknecht/P. Forstmoser/C.A. Zehnder, Verlag Schulthess
- [Reu93] Christoph Reuss:
Analyse von Gegenstand und Umfang des Programmschutzes
Semesterarbeit bei Prof. Dr. C. A. Zehnder, Betreuer Paul Trunz
- [Höhn] Dr. Jakob Höhn
Die Vertragsbeziehungen zwischen EDV-Anbieter und EDV-
Anwender bei der gemeinsamen Entwicklung von
Branchensoftware
Reihe Computer und Recht, Band 19,
Hrsg. K. Bauknecht/P. Forstmoser/C.A. Zehnder, Verlag Schulthess

- [Lutz93] Dr. jur. Martin J. Lutz:
Der Schutz der Computerprogramme in der Schweiz
Fundstelle: GRUR International 9/93 S. 653-663
- [CAZ94] Prof. Dr. Carl August Zehnder
Rechtsschutz von Computerprogrammen
Das neue Urheberrecht aus der Sicht der Informatiker
Fundstelle: Informatik 2/94 (Verbandsorgan der Gesellschaft
Schweizer Informatiker (SI) und des Wirtschafts-
informatik-Fachverbands (WIF))

9.3 Literatur zum Softwareschutz im Ausland

- [Kön92] Dr. M. Michael König:
Rechtsschutz von Computerprogrammen: Europäisches Recht mit fragwürdigen Bestimmungen schützt Programme
Fundstelle: c't 12/92, S. 100
- [Kön93] Dr. M. Michael König:
EG-Recht: Urheberrechtsschutz für Computerprogramme
Zur 2. Änderung des deutschen Urheberrechtsgesetzes vom 24.6.93
Fundstelle: c't 10/93:, S 58
- [Oman92] Ralph Oman, Register of Copyrights, Washington D.C.:
Urheberrecht, Computerprogramme und Halbleiterchips in den USA
Fundstelle: GRUR International 12/92 S. 886-892

9.4 Weitere zitierte Stellen für allgemeine Definitionen

- [CAZ91] Prof. Carl August Zehnder:
Informatikprojektentwicklung
2. Auflage, vdf 1991
- [Rau88] Dr. iur. Georg Rauber:
Der urheberrechtliche Schutz von Computerprogrammen
Reihe Computer und Recht, Band 17,
Hrsg. K. Bauknecht/P. Forstmoser/C.A. Zehnder, Verlag Schulthess