

Nutzung von Open Source Software in proprietären Softwareprojekten - eine Analyse aus urheberrechtlicher Sicht

Roman Heidinger

Abteilung für Informationsrecht und Immaterialgüterrecht, Wirtschaftsuniversität Wien
Althanstrasse 39-45, 1090 Wien
Österreich
roman.heidinger@wu-wien.ac.at
www.infolaw.at

Abstract: Open Source Software kann ohne Zahlung von Lizenzgebühren eingesetzt werden. Jedoch zwingen Lizenzen, die – wie die General Public License (GPL) - auf dem „Copyleft“-Prinzip beruhen, den Entwickler dazu, Veränderungen an dieser Software wiederum mit Quellcode freizugeben. Dieser Beitrag analysiert, welche Möglichkeiten aus rechtlicher Sicht bestehen, Open Source Software im Rahmen von proprietären Softwareprojekten einzusetzen. Dabei wird vor allem auf die Nutzung GPL lizenzierter Programmbibliotheken eingegangen.

1. Einleitung

Aus betriebswirtschaftlicher Sicht erscheint es besonders erstrebenswert, Open Source Produkte im Rahmen von proprietären Softwareprojekten einzusetzen, da die Verwendung von Open Source Software kostenlos möglich ist. Die meisten Open Source Lizenzen – wie beispielsweise die General Public License (GPL)¹ – basieren jedoch auf dem so genannten „Copyleft“-Prinzip, wonach Veränderungen an Open Source Programmen nur wiederum unter den Bedingungen der entsprechenden Lizenz – d.h. lizenzkostenfrei und quelloffen – weiterverbreitet werden dürfen. Damit wird verhindert, dass geänderte Programme proprietär, also „unfrei“, vertrieben werden können.

Wer also beispielsweise GPL-lizenzierte Programme verbessert, muss diese Verbesserungen wiederum der Community in Quellcodeform zur Verfügung stellen². Das „Copyleft“-Prinzip betrifft jedoch nicht nur die Weiterentwicklung von GPL-Software selbst, sondern auch die Übernahme (direkte Integration) von GPL-lizenziertem Programmcode in eigene Applikationen. Dabei muss jeweils das entstehende Gesamtprogramm wiederum den Bedingungen der GPL unterstellt werden und zwar unabhängig davon, wie groß der übernommene Teil im Vergleich zur

¹ Eine deutsche Version der GPL ist unter <http://www.gnu.de/gpl-ger.orig.html> abrufbar (abgefragt am 20. Juni 2005).

² Der entsprechende § 2 b der GPL lautet dabei wie folgt: *„Sie müssen dafür sorgen, dass jede von Ihnen verbreitete oder veröffentlichte Arbeit, die ganz oder teilweise von dem Programm oder Teilen davon abgeleitet ist, Dritten gegenüber als Ganzes unter den Bedingungen dieser Lizenz ohne Lizenzgebühren zur Verfügung gestellt wird.“*

Eigenentwicklung ist. Dies bedeutet im Extremfall, dass schon die Übernahme einiger weniger Programmzeilen dazu führt, dass das gesamte, neu erstellte Programm der GPL unterstellt werden muss und somit nicht mehr proprietär vertrieben werden kann. Dieser Effekt, der zur Überleitung der GPL auf alle Programme führt, die auf GPL-geschütztem Code basieren, wird oftmals auch als „*viral effect*“ bezeichnet³. Aus diesen Folgen wird deutlich sichtbar, dass das „Copyleft“-Prinzip den Interessen proprietärer Softwarehersteller diametral entgegensteht.

Eine völlig problemlose direkte Integration von Open-Source-Programmen in proprietäre Softwareprojekte ist hingegen bei den so genannten „Non Copyleft“-Lizenzen, wie zum Beispiel der BSD-Lizenz⁴ möglich. BSD-lizenzierte Software kann - auch in veränderter Form - ohne Quelltext weiterverbreitet werden, es muss lediglich ein Copyright-Hinweis der ursprünglichen Entwickler sowie ein Haftungs- und Gewährleistungsausschluss beigefügt werden⁵. BSD-lizenzierte Software kann somit ohne Probleme in kommerzielle Produkte integriert werden. Allerdings ist zu betonen, dass nur ein geringer Anteil von Open Source Projekten auf „Non Copyleft“-Lizenzen beruht.

Da eine direkte Integration von Programmteilen aus Open Source Software in proprietäre Programme zumeist aufgrund des „Copyleft“-Prinzips nicht zulässig ist, gilt es zu untersuchen, ob eine entsprechende Integration auf indirektem Weg realisiert werden kann. Dabei bietet sich die Zusammenfassung von Routinen in Programmbibliotheken an. Die dahinter steckende Grundidee ist, dass die für die Entwicklung proprietärer Programme benötigten Funktionen aus bestehender Open Source Software in Programmbibliotheken integriert werden könnten und so technisch fast dieselbe Wirkung wie bei der direkten Integration des Programmcodes erreicht werden kann. In der Folge soll dieser Themenkomplex exemplarisch an den Bestimmungen der GPL dargestellt werden, da diese Lizenz ist mit einer Verbreitung von 69%⁶ die bei weitem bedeutendste Open Source Lizenz ist.

2. Rechtliche Analyse der Nutzung GPL lizenzierter Programmbibliotheken

Bei der Frage, ob es rechtlich zulässig ist, GPL-lizenzierte Programmbibliotheken im Rahmen proprietärer Programme zu nutzen, muss grundsätzlich zwischen statischer und dynamischer Verlinkung unterschieden werden.

Das **statische Verlinken** ist ein Vorgang, der während der Erstellung des Programms erfolgt, so dass der Benutzer ein fertig zusammengesetztes Programm erhält. Bei vollständig statisch verlinkten Programmen besteht das ausführbare Programm aus einer

³ Spindler, Rechtsfragen bei Open Source (2004), Otto Schmidt, Köln, 111.

⁴ BSD steht für Berkely Software Distribution.

⁵ Für weitere Details vgl. Arlt/Brinkel/Volkmann in Spindler, Rechtsfragen bei Open Source (2004), Otto Schmidt, Köln, 317ff.

⁶ Die Verbreitung der GPL wurde anhand von insgesamt 60.343 Open Source Projekten, die am 21. Februar 2005 auf der Plattform *SourceForge* gehostet waren, berechnet.

einzigsten Datei. Bei dieser Vorgangsweise bleibt der Quelltext der GPL-lizenzierten Bibliothek zwar unverändert, jedoch entsteht durch das Linken des zugreifenden Programms mit der Bibliothek ein dauerhaft verbundenes, neues Programm. Die dadurch entstehende ausführbare Datei wird zu einem abgeleiteten Programm der Bibliothek im Sinn von § 2 b GPL und muss somit quelloffen und ohne Lizenzgebühren vertrieben werden⁷. Aus diesem Grund wurde auch für Bibliotheken eine Abschwächung der GPL formuliert, die so genannte „Lesser-General-Public-License“ (LGPL). So lizenzierte Bibliotheken dürfen ohne weiteres zu proprietären Anwendungen dazugelinkt werden, ohne dass das Endprodukt wiederum automatisch unter die Bedingungen der LGPL fällt. Allerdings beträgt die Verbreitung der LGPL nur 12%⁸, sodass in den meisten Fällen die strengeren Regeln der GPL zu beachten sind.

Dynamische Verlinkung hingegen bedeutet, dass das ausführbare Programm und die Bibliothek, auf die zugegriffen wird, erst im Zeitpunkt der Ausführung des Programms verbunden werden. Beispiel für dynamisch verlinkte Bibliotheken sind zum Beispiel die unter Windows verwendeten Dynamic Link Libraries (DLL). Die Analyse, ob ein Programm, das Routinen aus einer dynamisch verlinkten GPL-Bibliothek aufruft, nur unter den Lizenzbedingungen der GPL vertrieben werden darf, muss sowohl aus urheberrechtlicher Sicht als auch anhand der Bestimmungen der GPL erfolgen.

Der erste Schritt der Analyse beruht auf der Überlegung, dass jene Handlungen, die aus Sicht des Urheberrechts zulässig sind, ohne Rücksicht auf etwaige einschränkende Bestimmungen der GPL durchgeführt werden können. Nach § 24 Abs 1 UrhG gilt, dass die Benutzung eines Werks bei der Schaffung eines anderen Werkes zulässig ist, wenn es im Vergleich zum benutzten Werk ein selbstständiges neues Werk darstellt (sog. **freie Benutzung**). An einer solchen Nachschöpfung besteht kein abhängiges, sondern ein selbstständiges Urheberrecht, zu dessen Verwertung es keiner Einwilligung des Urhebers des benutzten Werkes bedarf. Für die freie Benutzung ist kennzeichnend, dass trotz des Zusammenhangs mit einem anderen Werk ein selbstständiges Werk vorliegt. Angesichts der Eigenart des benutzenden Werkes müssen die Züge des benutzten Werks verblasen⁹. Dies trifft gerade für das Schreiben eines Programms zu, das eine dynamisch verlinkte Bibliothek nutzt. Das Programm bezieht sich zwar auf die GPL-lizenzierte Bibliothek, übernimmt jedoch keinesfalls Teile aus ihr. Es entsteht somit ein eigenes Werk, zu dessen Erstellung es nicht der Zustimmung des Urhebers bedarf¹⁰. Somit kann festgehalten werden, dass an Programmen, die auf eine Bibliothek im Wege der dynamischen Verlinkung zugreifen, ein eigenständiges Urheberrecht erworben wird. Ein solches Programm kann daher ohne Rücksichtnahme auf die Bedingungen der GPL vertrieben werden. Vor allem können aber für das neu erstellte Programm Lizenzgebühren erhoben werden.

⁷ Vgl. Metzger, Freiheit den Bibliotheken, Linux Magazin 03/2001.

⁸ Zur Erhebung siehe FN 6.

⁹ Die freie Benutzung setzt somit voraus, dass das fremde Werk nicht in identischer oder umgestalteter Form übernommen wird, sondern lediglich als Anregung für das eigene Werkschaffen dient (vgl. Loewenheim in Schricker, Urheberrecht², Rz. 10 zu § 24).

¹⁰ Grundsätzlich wäre auch das gemeinsame Ausführen beim Nutzer eine an die Zustimmung des Urhebers gebundene Handlung, allerdings wird das Ausführen von Programmen durch die GPL in keiner Weise beschränkt (vgl. § 0 GPL).

In einem zweiten Schritt muss nun noch geprüft werden, ob die gemeinsame Verbreitung des proprietären Programms mit der GPL-lizenzierten Bibliothek auf einem Datenträger zulässig ist. In diesem Zusammenhang muss § 2 Abs 2 der GPL beachtet werden, der folgendes festlegt: „Wenn identifizierbare Teile des Datenwerkes [Anm. gemeint ist ein Computerprogramm] nicht von dem Programm abgeleitet sind und vernünftigerweise als unabhängige und eigenständige Datenwerke für sich selbst zu betrachten sind, dann gelten diese Lizenz und ihre Bedingungen nicht für die betroffenen Teile, wenn Sie diese als eigenständige Datenwerke weitergeben. Wenn Sie jedoch dieselben Abschnitte als Teil eines Ganzen weitergeben, das ein auf dem Programm basierendes Datenwerk darstellt, dann muss die Weitergabe des Ganzen nach den Bedingungen dieser Lizenz erfolgen, deren Bedingungen für weitere Lizenznehmer somit auf das gesamte Ganze ausgedehnt werden – und somit auf jeden einzelnen Teil, unabhängig vom jeweiligen Autor“. Damit ist klar ersichtlich, dass der gemeinsame Vertrieb von GPL-lizenzierter Software mit anderen Programmen – auch wenn diese eigenständige Werke im Sinn des Urheberrechtsgesetzes sind – dann unzulässig ist, wenn die beiden Programme voneinander abhängig sind¹¹. Das Vorliegen eines eigenständigen Programms ist nur dann anzunehmen, wenn die hinzugefügten Bestandteile unabhängig vom vorbestehenden Code in unterschiedlichen Adressräumen ausgeführt werden¹². Genau dieses Kriterium ist jedoch bei dynamisch verlinkten Programmbibliotheken nicht erfüllt, da diese beim Ausführen mit dem aufrufenden Programm im Arbeitsspeicher des Rechners zusammengeführt werden und dort gemeinsam ausgeführt werden. Zusammenfassend kann daher festgestellt werden, dass ein proprietäres Programm, das auf eine Programmbibliothek im Wege der dynamischen Verlinkung zugreift, zwar ein eigenständiges Werk im Sinn des Urheberrechts ist, der gemeinsame Vertrieb beider Komponenten ist aber nicht mit den Lizenzbedingungen der GPL vereinbar.

Eine GPL-konforme Nutzung dynamisch verlinkter Programmbibliotheken ist daher nur dann möglich, wenn der Vertrieb der proprietären Programmdatei unabhängig von der Verbreitung der GPL-lizenzierten Bibliotheksdatei erfolgt. Ein mögliches Vertriebssystem könnte daher darauf abstellen, dass der Benutzer die benötigten Programmbibliotheken aus dem Internet bezieht und auf seinem Rechner mit dem proprietären Programm zusammenführt.

¹¹ Vgl. in diesem Zusammenhang auch § 2 Abs 4 GPL.

¹² Vgl. Jaeger/Metzger, Open Source Software (2002), C.H. Beck, München, S. 42.

3. Fazit

Die Nutzung von Programmbestandteilen aus Open Source Software in proprietären Softwareprojekten ist nur in einem sehr eingeschränkten Ausmaß möglich.

Die direkte Integration von GPL-lizenziertem Programmcode sowie die Nutzung von Programmbibliotheken auf dem Weg der statischen Verlinkung ist in proprietären Softwareprojekten nicht möglich, da das entstehende Resultat wieder quelloffen vertrieben werden müsste. Eine Nutzung von Bestandteilen GPL-lizenzierter Software kann lediglich dann erfolgen, wenn der entsprechende Programmcode in der Form einer Programmbibliothek mit dem proprietären Programm auf dem Wege der dynamischen Verlinkung verbunden wird. Dabei muss jedoch die Verbreitung der Bibliotheksdatei getrennt vom proprietären Programm erfolgen.