Copyright or copyleft?
An analysis of property regimes for software development

Paul B. de Laat *

Faculty of Philosophy, University of Groningen, Oude Boteringestraat 52, 9712 GL Groningen, The Netherlands

Abstract

Two property regimes for software development may be distinguished. Within corporations, on the one hand, a Private Regime obtains which excludes all outsiders from access to a firm's software assets. It is shown how the protective instruments of secrecy and both copyright and patent have been strengthened considerably during the last two decades. On the other, a Public Regime among hackers may be distinguished, initiated by individuals, organizations or firms, in which source code is freely exchanged. It is argued that copyright is put to novel use here: claiming their rights, authors write 'open source licenses' that allow public usage of the code, while at the same time regulating the inclusion of users. A 'regulated commons' is created. The analysis focuses successively on the most important open source licenses to emerge, the problem of possible incompatibility between them (especially as far as the dominant General Public License is concerned), and the fragmentation into several user communities that may result.
© 2005 Elsevier B.V. All rights reserved.

Keywords: Commons; Intellectual property rights; Licensing; Open source; Software

1. Introduction

In the 'new economics of science', the production and distribution of knowledge are analysed from the point of view of information disclosure (cf. Dasgupta

and David, 1994). The central question is whether knowledge is pursued in order to increase the public stock of knowledge, or to generate rents from its private exploitation. From this perspective, two kinds of systems may be distinguished, usually referred to as *Science* and *Technology*. In the former, knowledge is to be published openly, while in the latter, results are to remain a secret. This distinction between regimes can also be arrived at by focusing upon market mechanisms. Technology is the realm of the market, supported by intellectual property rights (IPRs) as granted by the state. Science, on the other hand, is a regime created

by the state in an effort to correct market failure by granting subsidies and creating public laboratories. As Dasgupta and David stress, knowledge workers may be 'scientists' or 'technologists'—or both. It is not their cognitive practices, but the regime in which they work that will decide upon the matter.

In this article, a specific kind of knowledge will be analysed: software. Under what kind of market and non-market modes are computer programs being developed? In line with the analysis above the distinction between public disclosure and private appropriation of knowledge will be the central focus. IPRs will feature prominently in the analysis, surprisingly in *both* regimes. In the case of software, the market regime is in force within companies, mostly producers of hardware or software. The non-market regime, somewhat unusually, obtains within communities of computer hackers, who can be found anywhere, both inside *and* outside universities. Therefore, in order to avoid misleading connotations, these two software regimes will no longer be referred to as Technology and Science, but as *Private* and *Public Regimes*, respectively.

Software merits special attention, as it has unique qualities that set it apart from other types of knowledge products. A programmer starts with an idea that is specified in an algorithm. It is this algorithm that can be programmed. In the early days of computing, this took place directly in a language that the computer could read (machine language). Such languages, however, are difficult to 'read' by human beings, let alone to change. Therefore, higher order languages, more easily readable, were invented to make programming easier. Since then, programming is done in a computer language, which subsequently has to be transformed into a machine language. In computer jargon: *source code* has to be translated by a compiler into *object code*. It will become clear that this distinction between algorithm, source code and object code plays a vital role in both regimes.

First the Private Regime as it obtains within companies will be analysed. It will be shown that from the 1980s onwards, both secrecy and IPRs have evolved considerably. Next, the analysis focuses on the Public Regime of hackers freely sharing source code, which has evolved alongside. IPRs will be shown to play a rather different role here: as copyright holders, authors created new kinds of licenses that regulate the inclusion of others (instead of excluding them). From the 1990s

onwards, this movement for 'open source software' has also made inroads into the private sector: by way of experiment, some firms took a free ride on existing projects, or opened up software projects of their own to outside hackers. The analysis will show that, as a result, open as well as mixed property regimes evolved (combining elements of both the Private and the Public Model), and new open source licenses were formulated in order to accommodate business interests.

Within the academic community, open source software development is increasingly attracting attention. For purposes of comparison, the following sources should be mentioned in particular. First and foremost several studies by Yochai Benkler and Lawrence Lessig, legal scholars who opened up valuable avenues for research on open source software (Benkler, 2001, 2002a,b; Lessig, 2002a,b). More recently, Research Policy published a special issue on this matter (vol. 32, no. 7, 2003), in which several authors touched upon the subject of property rights. From these, West (2003) in particular is relevant for purposes of comparison. All of these sources I will refer to later at several instances. On beforehand, it would seem useful to mention on what account my analysis is different. Broadly speaking, concerning the matter of IPRs, I do not restrict myself to the simplified picture of the open source community as using only two types of license (either the General Public License or the Berkeley Software Distribution license; cf. below), but I explore more fully the whole range of 'open source licenses' that evolved, whether drafted by individuals, organizations or companies. Moreover, the complexities arising from combining source code with different licenses are explored. As a result, a more fine-grained picture will emerge of the open source movement and its communities of users.

2. Private Regime

2.1. Protection of intellectual property

In order to develop new products and/or processes, companies have to invest in research and development (R&D). These investments, however, are tricky: the fruits of them may easily be expropriated and/or imitated by competitors. Companies, therefore, have no choice but to protect their competitive advantage from