Research on Legal Issues of Computer Software Reverse Engineering

Jing Xu 1 *, Zhanfeng Li 1

1 Yantai Vocational College, Yantai City, Shandong Province, China, 264670

*Corresponding author e-mail: lizhanfeng@ytvc.com.cn

Abstract. With the development of computers, software has been applied in all walks of life. With the development of software, a lot of problems and controversies have appeared. For example, the emergence of software reverse engineering. Some people think that it has promoted the innovation and development of the software industry, and it has a great significance when promoting the development of science and technology society, but some people think that there are still some problems with the legality of reverse engineering. Therefore, it is necessary for us to learn from the advanced foreign legislation and make a detailed analysis of the conditions for the implementation of reverse engineering in our country.

Keywords: Software, Reverse Engineering, Intellectual Property Rights

1. Introduction

At present, software industry is showing a rapid growth trend in our country. It has strategic significance if we use reverse engineering to enhance the industry's innovation capabilities. At the same time, it plays an active role in driving innovation, promoting technological development, and stimulating competition. In today's society, the understanding of software reverse engineering should be raised to a new height. There are still many gaps in current laws and regulations on reverse engineering. So we must learn from advanced foreign legislation to analyze the constituent elements of reverse engineering in our country. We need to improve relevant legislation as soon as possible [1].

2. Computer software reverse engineering related concepts analysis

2.1. Computer software

Computer software consists of two parts: one is a computer program, and the other is a document that explains how to use the program. The "program" here refers to a sequence of instructions expressed in code. A program is a set of instructions. We use this set of instructions to control the computer to make it realize the actions to be completed in the future. To put it vividly, a program is like a music score (program) written in Chinese (programming language). ), used to instruct people who can read Chinese and who are familiar with music scores to play this song.
2.2. Source program and target program
A program is a sequence of instructions that instructs a computer's actions, so how is a computer program generated? First of all, the program is usually written in a programming language, and the source program refers to a readable text document that is written in a certain programming language and has not been compiled [2]. However, the computer cannot directly accept and execute the source program written in high-level language. When the source program is input into the computer, it must be translated into a target program in machine language through a "translation program" before the computer can recognize and execute it. This "translation" process is called compilation. After compilation, the source program expressed in computer language, that is, the binary instructions represented by "0" and "1" that the computer can execute, is the target program. Under normal circumstances, it is believed that the source program and the target program are completely equal in substance, but there is only a difference in form. The source program or source code is one of the core of the software, and it is also the key target for reverse engineering [3].

2.3. Software compatibility
Software compatibility is often used as an important parameter to judge the pros and cons of software. Compatibility refers to a set of performances related to the adaptability of software in different systems. Compatibility is produced in an environment of free competition. This performance greatly reduces the cost of software development, which not only saves resources, but also makes it more convenient for users to operate and use. Therefore, it has become a common path for computer manufacturing and software industries. As early as the 1991 "Computer Program Legal Protection Directive", the European Union stated its position on whether reverse engineering is allowed to develop competitive compatible products: Reverse engineering is allowed as long as it is for the purpose of achieving compatibility. This classic exposition of "compatibility" also has an equally important reference for my country.

2.4. Reverse engineering of software
Reverse engineering, as the name suggests, is opposite to forward engineering, which refers to the reverse derivation of the normal product manufacturing process, and is a process of deriving the manufacturing principles of existing products. In our judicial interpretation, reverse engineering is interpreted as the use of certain technical means to separate, speculate, and restore the product obtained from legal means to grasp the core technical information of the product [4]. Software reverse engineering refers to the application of reverse engineering methods to the field of computer software. It is a work of reverse research on the target program of the software. Under the dominance of this kind of reverse thinking, the subject of reverse engineering starts with the target program and derives the basic elements of software development by others.

3. The process and purpose of computer software reverse engineering

3.1. The positive process of software development
The development process of computer software is a process in which high-level languages are compiled and converted into executable files. Before developing a software, you must first understand and investigate the market and user's ideas, and then perform a general design of the software system on the basis of demand analysis, that is, system design. After the overall design of the system, the developer needs to carry out the detailed design of the software system. After that, it enters the coding stage. After the compiled source code is assembled and compiled, the target program code and supporting documentation manuals that run smoothly and continuously can finally be generated. Through the above expression, we can describe the general process of software development in sequence: The forward process diagram of software development is shown in Figure 1 [5].
3.2. Process description of software reverse engineering
Contrary to forward engineering, in reverse engineering, the developer performs reverse interpretation, analysis, research and reasoning on the target program code of other people’s software, finds the source code, and tries to derive the basic components of other people’s software development, and then go through improve or upgrade to develop new products similar to or even better than the original software. In short, the software development process starts with requirements analysis to obtain the final target code and documentation; the opposite reverse engineering uses the target program or document as the entry point to trace back the source to obtain the original design of the software [6].

Core elements. The reverse process diagram of software development is shown in Figure 2.

Figure 1. Forward process diagram of software development

Figure 2. Reverse process diagram of software development

4. Legal problems in the implementation of reverse engineering of computer software
4.1. Copyright issues caused by reverse engineering of computer software
1) Discussion on the copyright protection mode and patentability of computer software As a functional work, the multiple expressions of software make the copyright law unable to provide comprehensive protection, because for software, the core value is not as simple as the surface code, but hidden behind the code. Design ideas. This provides clues for the transition from copyright to patent, forcing us to face a bold question: beyond the protection of copyright law, can we seek the patent system to provide more comprehensive and reliable protection for software? The answer is affirmative and clear. Intellectual achievements in software can also be protected by the patent system. The conflict between
the basic principle of "the dichotomy of thought and expression" of copyright and the special functional product of software cannot be reconciled, and the patent protection of software is overwhelming [7]. Although my country's patent law has not yet included computer software in its protection category, it is undeniable that the trend of software protection patentability will eventually spur and promote the improvement of legislation, and the future software protection model is bound to continue to expand and develop.

2) Computer software reverse engineering: fair use or copyright infringement No legislator can create laws based on a single factor. When making value choices, we must make judgments and choices based on the purpose and effectiveness of the legislation. For my country's intellectual property legislation, the ultimate goal and foothold of legislators must be based on the consideration of social public welfare to protect intellectual achievements, and the advocacy of legislation should be aimed at promoting technological innovation and progress and benefiting the public interest. When judging whether copying and its extension are conducive to the purpose of promoting the development of science and technology, it should be consistent with the legal value system of our country. Therefore, the copying behavior in software reverse engineering should not be included in the category of copy right in copyright law. Of course, in actual judgment, attention should also be paid to the balance of interests of all parties, and intellectual achievements should be evenly distributed under the common goal of scientific, technological and cultural progress and social benefits [8].

4.2. Technical secrets caused by reverse engineering of computer software

1) Commercial secret protection model of computer software Trade secrets refer to a type of technical and business information, which is not known to the public except for the rights holders, and its practicality brings economic benefits to the rights holders. For this reason, the rights holders use relevant confidential measures to ensure their safety. The source code in computer software is protected because it meets the constitutive requirements of trade secrets. With the development of software technology, the defects of copyright protection have become more and more prominent. A core value of computer software It is the creativity and ideas that are hidden under the external form and embody personal wisdom, while copyright only protects expression, not ideas. This makes the copyright protection model unable to effectively protect the interests of software rights holders. People turn to seek protection of trade secrets. Mode to keep the core idea of software technology secret. This method is a way to isolate the outside world from accessing core information by using its own confidential means. However, after the software product is publicly sold, the trade secret protection model bears the risk of being self-evident-once the legal holder of the software learns the secret through reverse engineering and other means, the previous secret may be revealed in In the public's field of vision, it is clear to the world. Then, the owner of a trade secret no longer has the exclusivity and exclusivity of the secret. It can be seen that the protection model of trade secrets cannot effectively and long-term protect the interests of software developers[9].

2) Computer software reverse engineering: fair competition or infringement of trade secrets Reverse engineering is an effective way to prevent owners of trade secrets from permanently possessing useful information. To a certain extent, it promotes that trade secrets are not monopolized by right holders but can be used by multiple parties, and achieves a balance between information generation and dissemination. Therefore, reverse engineering is not an act of infringing on trade secrets, and it should be regarded as a legal and fair competition.

4.3. Contract legal issues caused by computer software reverse engineering

1) Computer software license contract The significance of software reverse engineering is that, as a benefit coordination tool, it promotes the redistribution of intellectual property rights between software right holders and users. Regarding software rights holders, they continue to try to open up a variety of new protection methods and methods to make up for the shortcomings of copyright law protection. As a result, contract law has entered the vision of software right holders. This kind of software protection method is quite special, especially when the software developer discovers that the customer can also
become the subject of reverse engineering. Therefore, the software right holder has added many prohibitions to the software license contract to require the user not to implement reverse engineering. First of all, we must first understand the types of software license contracts. During the installation and use of many software, we often see software license contracts. There are two types of general commercial software license contracts: The first is an unpacking contract. This kind of contract is generally obviously set at the top of the software package, and users have only two choices: acceptance and rejection, and no other options. Opening the package means that the software user has agreed to the contract content by default. The second is to click on the contract. When the user installs the computer software, a plug-in will pop up to ask the user for further operations. The plug-in appears in front of the user in the form of a button, of course, the user can only choose to agree to the button, otherwise the installation will be interrupted.

2) Software reverse engineering prohibition clause: autonomy of will or overlord clause Article 4 of my country’s "Contract Law" stipulates: "The parties enjoy the right to voluntarily enter into contracts in accordance with the law, and no unit or individual may illegally interfere." The essence of the contract under the legal framework reflects the high degree of autonomy of the parties. Any unit or individual has no right to interfere with the choice made by the parties after weighing various interests. However, in actual transactions, merchants take advantage of the asymmetry of information held by consumers and add unreasonable transaction conditions during transactions, which violates the corresponding provisions of my country's "Contract Law" and "Anti-Monopoly Law." In addition, according to Article 39 of my country’s "Contract Law", "standard clauses are clauses that are drawn up in advance by the parties for repeated use and have not been negotiated with the other party when the contract is concluded.” “Prohibited and counter clauses” are closer to the format in nature. Clauses, because the party who drafts the contract often uses its advantage of information asymmetry with consumers to formulate clauses that are obviously unfair in content[10]. This is why the law strictly restricts the effective elements of standard clauses in order to protect consumer rights, reduce transaction risks, and maintain market order.

5. Conclusion
In a word, reverse engineering is the beginning of all innovation, and no one can invent and create in a vacuum. The development of the software industry is more dependent on reverse engineering behavior. Judging from the reality of our country's software industry, we must continue to expand the ways and channels of information utilization and continue to improve independent innovation and development capabilities.

References
[1] Wu Handong. Research on the System of Fair Use of Copyright [M]. Beijing: China University of Political Science and Law Press.
[2] Wu Handong. Intellectual Property Law [M]. Beijing: Law Press.
[3] Zheng Secheng. Intellectual Property Theory [M]. Beijing: Social Sciences Academic Press, 2007. Beijing: China Founder Press, 2004. (in Chinese)
[4] Ying Ming. Copyright Issues in Reverse Engineering of Computer Software [M]. Beijing: Law Press, 2004.
[5] Ying Ming, Sun Yan. Copyright Protection of Computer Software [M]. Beijing: Intellectual Property Publishing House, 2009.
[6] Feng Xiaoqing. The Theory of Balance of Interest in Intellectual Property Rights [M]. Beijing: China University of Political Science and Law Press, 2006.
[7] Feng Xiaoqing. Philosophy of Intellectual Property Law [M]. Beijing: China People's Public Security University Press, 2003.
[8] Cao Wei. Reflection and Transcendence of Computer Software Copyright Protection [M]. Beijing: Law Press, 2010.
[9] Li Xiang. American Copyright Law: Principles, Cases and Materials [M]. Beijing: China
University of Political Science and Law Press.

[10] Lv yan, Ed. Research on intellectual property protection of computer software [M]. Beijing: law press, 2005.